RSX-11/3271
PROTOCOL EMULATOR
System Manager's Guide

Order No AA-C739C-TC

November 1982

SUPERSESSION/UPDATE INFORMATION:	This is a revised manual.
OPERATING SYSTEM AND VERSION:	RSX-11M V4.0 RSX-11M-PLUS V2.0
SOFTWARE VERSION:	RSX-11M/3271 Protocol Emulator V3.0

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Northboro, Massachusetts, 01532.

digital equipment corporation · maynard, massachusetts

First Printing, August 1978
First Revision, January 1981
Second Revision, November 1982

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

Copyright © 1978, 1981, 1982 by Digital Equipment Corporation

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	PDT
DATATRIEVE	TRAX	

Contents

Page

Preface ix

Chapter 1 The RSX-11/3271 PE: Features and Configuration Requirements

1.1 The 3271 PE Features . . . 1-1
1.2 The 3271 PE Communications and Environment . . . 1-4
1.3 Information Display System Characteristics Emulated by the PE . . . 1-5
1.4 Information Display System Characteristics Emulated by the TE . . . 1-6
1.5 Configuration Requirements . . . 1-7
1.5.1 Hardware Configuration for the RSX-11/3271 PE . . . 1-7
1.5.2 Software Requirements for the RSX-11/3271 PE . . . 1-9
1.5.3 Host Hardware and Software Requirements . . . 1-9

Chapter 2 PE Structure and Components

2.1 Overview of 3271 PE Architecture . . . 2-4
2.2 3271 PE Components . . . 2-5
2.2.1 Communications Executive (CEX) . . . 2-6
2.2.2 Communications Executive Processes . . . 2-7
2.2.2.1 LLC Processes . . . 2-7
2.2.2.2 DLC Processes . . . 2-8
2.2.2.3 DDM Processes . . . 2-9
2.2.3 Communication Pool (POOL..) . . . 2-11
2.2.4 Loading Initializing Tasks . . . 2-11
2.2.5 System Manager Utilities and the PE System Data Bases . . . 2-12
2.2.5.1 PE System Data Bases . . . 2-12
2.2.5.2 System Management Utilities . . . 2-13
2.2.6 System Maintenance Utilities . . . 2-14

Chapter 3 Pre-generation Requirements and Considerations

3.1 RSX-11M/M-PLUS Considerations . . . 3-2
3.1.1 Customizing the RSX Executive for PE Support . . . 3-2
3.1.2 Identifying Communications Devices . . . 3-4
3.1.3 Treatment of Parallel Questions in PEGEN and SYSGEN . . . 3-4
3.1.4 Loading Options . . . 3-5

3.2 Generation Requirements . . . 3-5
3.3 Partition Layout . . . 3-6
3.4 Buffer Considerations . . . 3-7
3.4.1 Allocation of Memory for Buffer Space . . . 3-8
3.4.2 Large Data Buffer (LDB) Requirements . . . 3-8
3.4.3 Communications Control Block (CCB) Requirements . . . 3-11
3.5 Assignments of Control Unit and Station Addresses . . . 3-12

Chapter 4 The PEGEN Generation Dialog

4.1 The PEGEN Dialog Display Format: Questions and Responses . . . 4-1
4.1.1 Questions . . . 4-2
4.1.2 Optional Explanatory Text . . . 4-2
4.1.3 Formats for Responses to the Questions . . . 4-2
4.1.4 Default Values Applied in Event of Non-responses . . . 4-3
4.2 PEGEN Structure . . . 4-3
4.3 Determining the Scope of the PEGEN Dialog . . . 4-6
4.4 Starting PEGEN: Distribution Kits and Considerations . . . 4-7
4.5 PEGEN Dialog . . . 4-10

Chapter 5 Installation and Verification

5.1 Installing, Loading, and Starting . . . 5-1
5.2 Post-Installation Verification . . . 5-2
5.2.1 Testing Line Software . . . 5-3
5.2.2 Testing Line Transmission/Reception Capability . . . 5-4
5.2.3 Verifying Proper Reception of Host Polls . . . 5-4

Chapter 6 Operating and Maintaining the Configuration

6.1 Loading and Unloading PE Software . . . 6-1
6.1.1 Loading the Communications Executive . . . 6-1
6.1.2 Unloading the Communications Executive . . . 6-2
6.1.3 Loading a Line (and/or Changing Controller Values) . . . 6-2
6.1.4 Unloading a Line . . . 6-5
6.1.5 Loading a Process . . . 6-5
6.1.6 Unloading a Process . . . 6-7
6.2 Start Up/Shut Down . . . 6-7
6.2.1 Start Up . . . 6-7
6.2.2 Shut Down . . . 6-8
6.3 Controlling Line State . . . 6-9
6.4 Monitoring Lines . . . 6-10
6.4.1 Monitoring Line Status . . . 6-10
6.4.2 Monitoring Line Activity . . . 6-11
6.4.2.1 Using NCP to Monitor Line Activity . . . 6-11
6.4.2.2 Using DCE to Monitor Line Activity . . . 6-12
6.5 Monitoring the System with NCP . . . 6-16
6.6 Modifying the Configuration . . . 6-18
6.6.1 Reloading/Renaming Process Partitions . . . 6-18
6.6.2 Adding Lines . . . 6-19
6.6.3 Changing Configuration Parameters . . . 6-19
6.6.3.1 Modifying a Generated Configuration Using NCP or VNP . . . 6-19
6.6.3.2 Modifying the Configuration Using CFE . . . 6-20
6.6.4 Modifying PE Controller and Station Addresses . . . 6-27
6.6.5 Reconfiguring with RSX-11M-PLUS . . . 6-29

Chapter 7 3271 Protocol Emulation, Message Formats, and Operation Sequences

7.1 Overview of 3271 Protocol Emulation and Communications 7-1
7.2 BISYNC Data-Link Control Characters 7-3
7.2.1 Equipment Control Characters 7-4
7.2.2 Message Framing Control Characters 7-5
7.2.3 Message Interchange Control Characters 7-6
7.2.3.1 Single Characters 7-7
7.2.3.2 Two-Character Control Sequences 7-8
7.2.4 BISYNC Messages 7-10
7.3 Message Formats and Operation Sequences 7-11
7.3.1 Polling 7-11
7.3.2 READ MODIFIED Operation 7-13
7.3.3 STATUS AND SENSE READ 7-16
7.3.4 Selection 7-18
7.3.5 WRITE Operation 7-21

Chapter 8 Error Handling and Troubleshooting

8.1 Error Handling 8-1
8.1.1 Installation and Reloading Errors 8-1
8.1.2 Runtime Communications Errors 8-2
8.1.3 Component Errors 8-3
8.2 PE Troubleshooting Tools 8-3
8.2.1 Using NCP 8-3
8.2.2 Using the Error Logging and Statistics Routine (DCE) 8-4
8.2.3 Using the Trace Program (TRA) 8-4
8.2.3.1 Trace Program Operation 8-4
8.2.3.2 Trace Program Commands 8-5
8.2.3.3 Trace Program Example 8-7
8.2.3.4 Interpreting Retrieved Messages 8-7
8.2.4 Loopback Test Progran. 8-11
8.2.4.1 LBK Commands 8-13
8.2.4.2 Sample LBK Session 8-16
8.2.4.3 How to Implement Your Own Loopback Test Program 8-18
8.2.5 Using the Network Dump Analyzer (NDA) to Analyze Crash Dumps 8-21
8.2.5.1 Obtaining a Crash Dump 8-22
8.2.5.2 Running NDA 8-23
8.2.5.3 NDA Command Strings 8-23
8.2.5.4 Indirect Command Files 8-25
8.2.5.5 Switches 8-26
8.3 Troubleshooting Procedures 8-29
8.3.1 Troubleshooting Software Problems 8-29
8.3.2 Troub'eshooting Hardware Problems 8-30
8.4 3271 Protocol Emulator Component Error Messages 8-32
8.4.1 CFE Error Messages 8-32
8.4.2 DCE Error Messages 8-34
8.4.3 DLX Error Messages and Codes 8-35
8.4.4 LBK Error Messages 8-36
8.4.5 NCP Error Messages 8-38
8.4.6 NDA Error Messages 8-44
8.4.7 NTL Error Messages 8-48
8.4.8 TRACE (TRA) Program Messages 8-54
8.4.9 VNP Error Messages 8-54

Appendix A Sample PEGEN Listings and Related Items

A.1 Sample PEGEN Listing A-1
A.2 Configuration File for 3271 PE A-13

Appendix B Partition Sizes

Appendix C Device Settings

Appendix D Configuration File Editor Utility (CFE)

D.1 Using the HELP Command D-1
D.2 Invoking CFE D-2
D.3 Exiting CFE D-3
D.4 CFE Dialog Example D-3
D.5 CFE Command Descriptions D-4

Appendix E Network Control Program Utility (NCP)

E.1 Using the HELP Command E-1
E.2 Invoking NCP E-2
E.3 Exiting NCP E-3
E.4 NCP Dialog Examples E-3
E.5 NCP Command Descriptions E-4

Appendix F Virtual Network Program Utility (VNP)

F.1 Using the HELP Command F-1
F.2 Invoking VNP F-2
F.3 Exiting VNP F-3
F.4 VNP Command Descriptions F-3

Figures

1-1 PDPs Linked to IBM Host 1-2
1-2 Conceptual View of a 3271 PE 1-3
2-1 Logical Subdivision of RSX-11/3271 PE System Components 2-2
2-2 3271 PE Lines and Pseudodevices 2-4
2-3 Relationships Between 3271 PE Components 2-6
3-1 Allocating the Communications Executive Partition 3-3
3-2 A 3271 PE Configuration with Two BU Controller Lines and Their Stations (Pseudodevices) 3-13
4-1 Divisions Within the PEGEN Process 4-5
5-1 Install, Load, and Start Procedure for PE: Sample Listing 5-2
6-1 DCE Error Counter Report 6-13
7-1 A BISYNC Message Stream 7-10
7-2 Operation Sequence for Poll 7-12
7-3 Operation Sequence for READ 7-15
7-4 Operation Sequence for Selection 7-20
7-5 Operation Sequence for WRITE 7-23
8-1 A Trace Program Report, Short Form (72-Character lines) 8-9
8-2 A Trace Program Report, Long Form (132-Character lines) 8-10
8-3 Systems Under Test 8-11
8-4 Line Loop Test 8-12
8-5 Modem Loop Test (Local) 8-12
8-6 Modem Loop Test (Remote) 8-13
8-7 Display Generated by HELP Command 8-16

Tables

1-1	Approximate Disk Space Needs for Target Disk	1-8
2-1	Component Name/Component Subgroup Cross-Reference	2-3
3-1	PE Buffers	3-10
3-2	LDBs Required for Devices Used by the PE	3-10
3-3	Controller and Station Address Translation Table	3-14
3-4	How TU Devices Are Assigned to PE Stations and TRA Units	3-16
4-1	Summary of PEGEN Responses	4-3
4-2	PEGEN Component Selection	4-6
6-1	Responses to Line Status Inquiry	6-10
6-2	3271 PE Error Counters	6-14
6-3	CETAB Values Changed by CFE	6-20
6-4	RSX-11M-PLUS Device Names	6-30
7-1	Functions and Hexadecimal Representations of BISYNC Equipment Control Characters	7-4
7-2	Functions and Hexadecimal Representations of BISYNC Message Framing Control Characters	7-6
7-3	Functions and Hexadecimal Representations of BISYNC Message Interchange Control Characters	7-9
7-4	Legal Polling Addresses for the 3271 Protocol Emulator	7-12
7-5	Status and Sense Byte Definitions for 3271 Protocol Emulator	7-17
7-6	Legal Selection Addresses for the 3271 Protocol Emulator	7-19
8-1	CSR Bit Settings for Modem Status Reads (DU11, DUP11, DUV11 line interfaces)	8-20
8-2	LSR Bit Settings for Modem Status Reads (DV line interfaces)	8-20
8-3	NDA File Default Values	8-24
8-4	NDA Analysis Control Switches	8-26
8-5	NDA Function Control Switches	8-27
C-1	DV11 Parameter Selection Switches	C-1
E-1	System Counters	E-20

Preface

This System Manager's Guide provides the information you will need to generate a 3271 Protocol Emulator (PE) to run on a RSX-11M or RSX-11M-PLUS system. In addition, this manual provides information for installing, operating, and maintaining the PE, and for troubleshooting and debugging PE-related problems.

This manual is both a tutorial for RSX system managers and a reference manual identifying components of configurations supported by the PE.

The new feature of the RSX-11/3271 PE is the support of the 3270 Terminal Emulator.

Chapter 1 introduces the 3271 PE, covering its functions and environment, and discussing the minimum hardware and software requirements for the RSX and host systems.

Chapter 2 describes the components of the 3271 PE, and their relationships.

Chapter 3 details pregeneration requirements and considerations.

Chapter 4 describes the procedure for generating the PE and lists all the displays that may appear during the generation process.

Chapter 5 explains how to install and verify your PE, and Chapter 6 explains how to operate and maintain it. These chapters explain how to load, control, monitor and modify PE software using any of several utilities. The appendixes provide command summaries for three of the major utilities used for these functions: the Configuration File Editor (CFE), the Network Control Program (NCP), and the Virtual Network Program (VNP).

Chapter 7 discusses the communications protocol used by PE and the message formats and sequences for 3271 operations.

Chapter 8 provides guidelines for troubleshooting and debugging communications problems, and includes the error messages that may be displayed during use of the utilities in conjunction with the 3271 PE.

The appendixes provides sample listings of generation dialogs, partition size requirements, device setting requirements, and the command references for the CFE, NCP and VNP utilities.

Associated Documents

RSX-11M/M-PLUS system generation procedures are described only to the extent necessary to assure PE compatibility with the operating system. Complete RSX-11M/M-PLUS system generation procedures are detailed in the *RSX-11M System Generation and Management Guide* and the *RSX-11M-PLUS System Generation and Management Guide*.

The *RSX-11/3271 Protocol Emulator Programmer's Guide* provides programming instructions and a glossary defining terms used in both PE manuals.

This manual assumes that IBM documentation meets the needs of those who are responsible for operating and maintaining the IBM configuration.

Documentation Conventions

The following conventions are used in this manual in discussions of PE commands and statements:

UPPERCASE	Uppercase letters indicate keywords that the user must include with the command. (You can abbreviate keywords to the first three letters of each keyword.)
lowercase italic	Lowercase italic letters indicate the type of information to be supplied by the user, not the actual user input. When one of these variables requires a description of more than one word, the words are hyphenated.
[ ]	Brackets indicate an optional field that need not be included by the user. (Fields not enclosed in brackets must be included.) In general, brackets are logical symbols only and should not be included in the input. However, brackets surrounding a UIC (User Identification Code) must be included by the user.
{ }	Braces indicate a choice of fields, one of which must be included by the user.
space	Spaces delineate items in the command. The user must input spaces where shown.
red	User input in examples is printed in red to distinguish it from system output.

⟨RET⟩	This symbol is used to signify a carriage return. Although not always shown in this manual, you must follow the last item of user input in a command by a carriage return.
n.	A number followed by a point is a decimal value.

Documentation Conventions for Host-Initiated 3270 Commands

All host-initiated 3270 commands are represented in capitals, for example:

READ MODIFIED.

Chapter 1
The RSX-11/3271 PE: Features and Configuration Requirements

The RSX-11/3271 Protocol Emulator (PE) is a software product that enables the 3270 Terminal Emulator (TE) and application programs in RSX systems to communicate interactively with application programs in remote host systems that support the IBM 3271 Binary Synchronous Communication (BISYNC) protocol. The 3271 PE resides in a PDP-11 processor that runs under the RSX-11M or M-PLUS operating system. The PE appears to the remote host as an IBM 3271 BISYNC control unit.

The IBM 3271 control unit is a non-programmable hardware device that is part of the IBM Information Display System (IDS), a data communications subsystem. The IBM 3271 controls the operation of up to 32 input/output devices (printers and display terminals). It is a slave to the host: it cannot transmit or receive data without a directive from the host computer. (The IBM 3270 IDS system is discussed further in Section 1.3.)

1.1 The 3271 PE Features

The RSX-11/3271 Protocol Emulator performs a subset of the regular IBM 3271 control unit functions. It communicates only in response to host system directives. The RSX-11/3271 PE supports up to six synchronous communications lines, each of which is viewed by the host as a distinct 3271 control unit. Furthermore, each emulated control unit can support up to 32 pseudo input/output devices. The pseudodevices are copies of the 3270 Terminal Emulator software or RSX user-written applications programs.

NOTE

The maximum number of lines supported by the RSX-11/3271 PE depends on the processing power and configuration of your system. The maximum number of application programs that can be associated with a line depends on the Dynamic Storage Region (DSR) available.

The 3271 PE resides in a PDP-11 computer system that is capable of running applications and providing facilities for program development.

The PE can share a line with real 3271s and other 3271 PEs in a multipoint configuration but it cannot act as a host for 3271 I/O devices. Figure 1-1 shows 3271 PEs in point-to-point and multipoint connections with an IBM host.

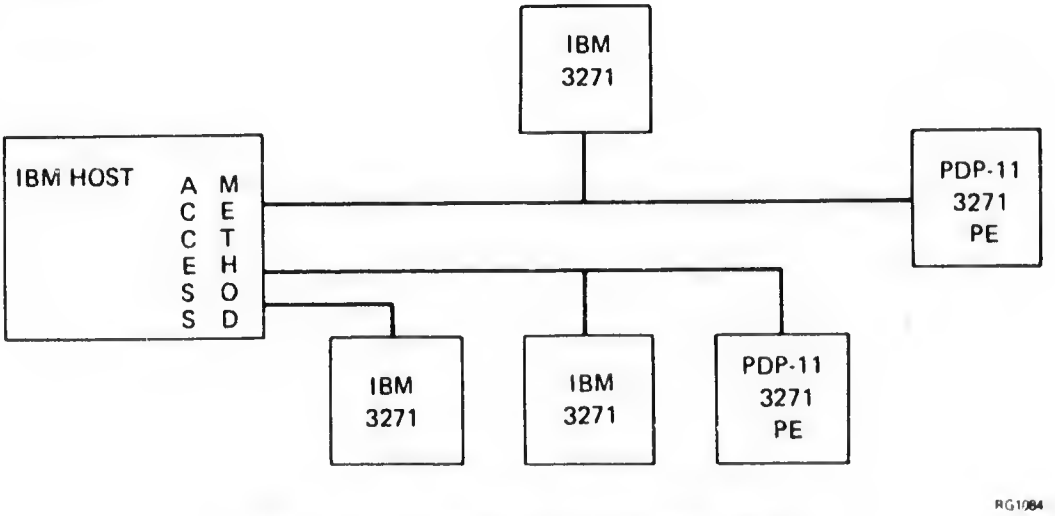


Figure 1-1: PDPs Linked to IBM Host

The PE communications protocol is compatible with standard IBM software used with the IBM 3270 Information Display System. The PE uses the 3270 IDS multipoint BISYNC Data Link protocol across the communications link and the 3270 IDS interface to the host computer system. The 3270 Terminal Emulator software and the RSX application programs associated with the 3271 PE appear to be real devices in the IBM 3270 family of output devices. Figure 1-2 illustrates this logical relationship for a 3271 PE/3270 TE system. The devices are considered as logical devices.

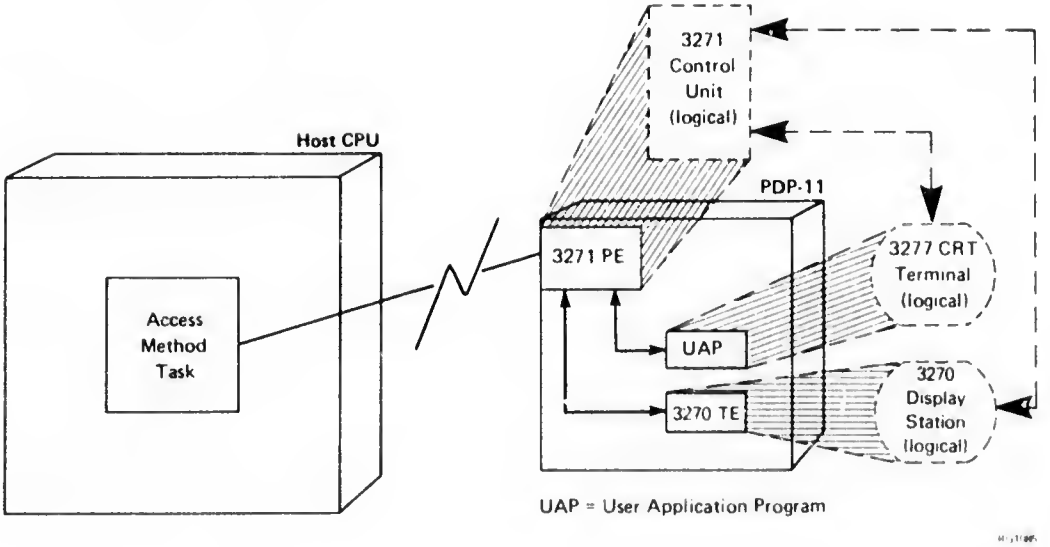


Figure 1-2: Conceptual View of a 3271 PE

The functions supported by the 3271 PE allow the 3270 TE and the RSX user application program to interact with the host to obtain or transmit data. The 3270 TE and the RSX user application program in conjunction with a complementary program in the host can perform the following operations:

- Make record inquiries and modifications to host or RSX data bases
- Update records in host data bases with data from the RSX system
- Transmit RSX-11M/M-PLUS data files to the host
- Receive host data files on a RSX-11M/M-PLUS system
- Update data bases on a PDP-11 by means of a program on a host system

The functions that the 3270 TE and the RSX application programs can offload from the host system include transaction spooling, output device formatting, data validity checking, and data base management. Thus, the communications link provided by the 3271 PE enables you to distribute processing. An application program can use the full power of the PDP-11 system to reduce host processor and communications facility loads, and at the same time exploit the host system's facilities for handling centralized tasks.

1.2 The 3271 PE Communications and Environment

In interacting between the RSX and host systems, the PE controls, at its end of the communications link, the physical processing of I/O requests. When the 3271 PE receives data in a message block from the host, it examines the message block header for the station address, determines the associated Terminal Unit (TU) pseudodevice, and passes the data to the 3270 TE or to the RSX application program logically attached to the TU pseudodevice. The TU driver is a pseudodevice that handles the interface with the TE and the application programs. See the discussion of the TU driver process (TUD) in Chapter 2 (Section 2.2.2.1).

When the 3271 PE receives data from the 3270 TE or from the RSX-11M/M-PLUS application program, it passes the data to the IBM host in the form of a message block. To the host, the message block appears to have been transmitted from a display station. (Protocol and message format are discussed in Chapter 7. Addressing is discussed in Chapter 3. Logical interaction between the RSX and host application programs is discussed in the *RSX-11/3271 Protocol Emulator Programmer's Guide.*)

The PE operates as a PDP-11 multiple device handler. It maintains and supervises the IBM BISYNC protocol over full- or half-duplex data links, in either point-to-point or multipoint configurations. The PE transmits and interprets the EBCDIC transmission code, and can communicate at transmission speeds of up to 9600 bits per second. However, line efficiency and throughput depend on the user application program and other tasks in the RSX system, and on the host communications system.

The 3271 PE can run on a mapped RSX-11M system or an RSX 11M-PLUS system. The PE uses the Communications Executive to support a user-level driver (Terminal Unit driver), a protocol emulation module (BSD), and device drivers. (Chapter 2 describes 3271 PE components.)

Optionally, the 3271 PE can support the KMC11-A microprocessor. This option improves throughput and offloads from the PDP-11 processor the character processing from those DUP synchronous lines controlled by the KMC.

The main characteristics of the RSX-11/3271 PE are:

- It operates on either RSX-11M or RSX-11M-PLUS systems, linking tasks to a host.
- It operates in point-to-point and multipoint line configurations.
- It communicates with systems that support 3271 Multipoint BISYNC Data Link Control protocol.
- It supports full- or half-duplex communications facilities.

- It optionally provides CPU offloading with a KMC-11 microprocessor option.
- It supports the 3270 Terminal Emulator software.
- It supports up to six synchronous communications lines, connecting with one to six host systems.
- It associates up to 32 PDP-11 tasks with each synchronous line. The tasks can be copies of the 3270 TE software.
- It supports a maximum line speed of 9600 baud.
- It supports the KG11 communication arithmetic option for the Cyclic Redundancy Check routine (CRC) calculation.

1.3 Information Display System Characteristics Emulated by the PE

The IBM 3270 Information Display System (IDS) is a fixed-function (non-programmable), terminal-oriented system designed for use in an on-line, interactive data communications environment. Hardware for the 3270 display system includes a variety of printers, display terminals, and control units.

The RSX-11/3271 PE supports the 3270 Terminal Emulator (TE) and user-written application programs which may emulate the functions and device characteristics of IBM 3277 display stations and IBM 3284 and 3286 printers. Two 3277 display station models may be supported by the 3271 PE: Model 1 is equipped with a 480 character buffer display capability, while Model 2 has a 1920 character buffer display capability. The 3270 Terminal Emulator packaged with the 3271 PE has a 1920 character buffer display capability. (Refer to Section 1.4 and to the *RSX-11/3271 Protocol Emulator Programmer's Guide* for a list of the specific features of the IBM products supported by the 3270 TE.)

The IBM 3270 IDS is always the slave station in relationship to the host computer station (master). The 3270 cannot transmit data unless polled (queried) by the associated host, and it cannot receive data unless selected by the host.

The IBM display system uses either the point-to-point or multipoint BISYNC protocol. The 3271 control unit and 3271 PE use only the multipoint protocol. This protocol can also be used for point-to-point configurations. For more information about BISYNC, see Chapter 7.

IBM host systems supported by the RSX-11/3271 Protocol Emulator are those that use EBCDIC code; RSX-11 systems use ASCII. To resolve this disparity, the RSX-11/3271 PE automatically translates data to be sent to the IBM host from ASCII to EBCDIC, and translates EBCDIC data received from the IBM host to ASCII.

1.4 Information Display System Characteristics Emulated by the TE

The RSX-11 3270 Terminal Emulator software uses a VT100 with the advanced video option. It supports the following features of the IBM 3270 IDS:

- IBM 3270 *specify features* that supply the 94-character EBCDIC character set and the EBCDIC transmission code
- IBM 3270 *special features* that supply the typewriter keyboard with 12 Program Function (PF) keys, the audible alarm, and the keyboard numeric lock
- 1920-character screen
- Field attribute characters which are not displayed but which provide the IBM programmer with information about the type of data that he can enter into the field that follows
- Protected and unprotected fields
- Alphanumeric or numeric data
- Intensified display, display, or nondisplay characters
- Mappings of the VT100 keys to the functions described in Chapters 5 and 6 of the *RSX-11/3271 Protocol Emulator Programmer's Guide*

The 3270 TE does not support the following features of the IBM 3270 IDS:

- 480-character screen
- Selector pen, security lock, and operator identification card reader
- Display field attribute: selector pen detectable or nondetectable
- Printer support including printer orders NL, CR, FF, and EM, and the COPY command
- Keyboards other than the typewriter keyboard
- TAB and NEW LINE as auto-repeat keys
- 3279 Display Station color support
- The 25th line on a 3278 Display Station

The 25th line usually supplies operator information. The 3271 TE supplies the following information instead:

- LED 1 and LED 2 on the VT100, labeled INS and INHIB, supply some of this information.
- Symbols concerning the hardware status of the communication line are replaced by error messages that appear on your screen.
- Symbols indicating that you have tried to enter data into protected fields or that you have tried to enter an incorrect data type into another field are unnecessary because the terminal's bell rings and the keyboard locks when you try to do this.
- Other symbols concern the printer, the security keylock, and the operator identification card reader which the TE does not support.

A VT100 without the advanced video option will not show reverse video and bold intensity.

1.5 Configuration Requirements

This section specifies options and describes the minimum hardware and software requirements for the 3271 PE and the host system.

1.5.1 Hardware Configuration for the RSX-11/3271 PE

The minimum hardware configuration needed to support the 3271 PE running the 3270 TE software is outlined as follows:

- A valid mapped RSX-11M or RSX-11M-PLUS configuration residing on any PDP-11 system supporting extended addressing. Refer to Appendix B for memory sizes.
- An additional 16K bytes of memory for the PE.
- A synchronous line interface: either a DV11, DUP11, DU11, or DUV11.
- Modems at each end of the communications line. The modems can be, for example, Bell 201-B, 201-C, 208-A, or the equivalent. If the line used is point-to-point, the modems should be strapped for 'continuous request to send' to avoid turn-around delays. When using OEM modems, make sure they guarantee equivalence.

NOTE

You must have a modem or modem eliminator, even if the remote system is nearby. Modems or modem eliminators provide a clock source that is not provided by the devices at the local or remote sites.

- VT100 terminals with advanced video option, attached to the RSX-11M/M-PLUS system.

When building a PE system, keep in mind that the target system must have room for the distribution kit, which is approximately 4800 blocks. The minimum amount of free disk space required for the generated system is approximately 2820 blocks. These restrictions apply only when building a PE system and do not apply to runtime configurations.

Table 1-1 gives the approximate disk space needs for the target disk. These figures assume the selection of all possible options and can be reduced by selecting fewer options and devices. They do not include the distribution kit files.

The minimum configuration of peripheral devices required to operate a 3271 PE/3270 TE system is a valid RSX-11M or RSX-11M-PLUS configuration and a DV11 or DUP11 synchronous line interface. You can add other storage and terminal devices to suit the needs of your applications.

Table 1-1: Approximate Disk Space Needs for Target Disk

PE components	
Disk resident	200 blocks
Needed during PE generation	120 blocks
Maps and lists	200 blocks
TE tasks	63 blocks
Total	583 blocks
Communications Executive and other System Utilities	
Disk resident	1100 blocks
Needed during PE generation	150 blocks
Maps and lists	950 blocks
Total	2200 blocks
PE with Communications Executive and Utilities	
Disk resident	1300 blocks
Needed during PE generation	370 blocks
Maps and lists	1150 blocks
TE tasks	63 blocks
Total	2883 blocks

Up to 32 logical devices or copies of the 3270 TE software per line may be defined for the 3271 PE. The number of logical devices supported depends on host system requirements as well as on any restrictions imposed by the RSX-11M or M-PLUS executive, such as the number of Device Control Blocks (DCBs), Unit Control Blocks (UCBs), and the memory requirements imposed by Communications Control Blocks (CCBs) and line tables.

Optional Hardware

The following hardware is optional:

- Two K words of additional memory for each additional synchronous line (beyond the first line).
- A KG11 arithmetic device for use with DU11 or DUP11 interfaces.
- A KMC11-A microprocessor for use with DUP11 interfaces. The KMC-11A is a general purpose UNIBUS microprocessor. This version of the RSX-11/3271 PE supports the KMC processor in combination with DUP-11s as a performance enhancement to the PDP-11 system. The KMC processor can control up to six DUP-11s, presenting a DMA interface to the 3271 PE. The driver that supports the KMC-DUP combination is named the KBU driver.

1.5.2 Software Requirements for the RSX-11/3271 PE

The 3271 PE runs on a mapped RSX-11M operating system (Version 3.2 or later) or on a RSX-11M-PLUS operating system (Version 1.0 or later). The protocol emulator requires that the following options be included in the RSX-11M or M-PLUS operating system:

- Loadable device drivers
- A partition in the RSX-11M executive of 5000 (octal) to 6000 (octal) bytes
- Sufficient pool space to allocate the device tables and CCBs

1.5.3 Host Hardware and Software Requirements

The host system must be a hardware system that supports the 3271 communications protocol. The software discussed is IBM software. Host systems made by other vendors must use equivalent software.

The RSX-11/3271 PE does not require any changes to the host operating system or network software. It can be defined to an IBM system configuration, for example, by means of normal system generation procedures for installing an IBM 3271 control unit (by adding a controller on a line, or a line on a controller).

A variety of IBM operating systems, access methods, application programming packages, and language compilers and assemblers can be used to support a communications link that involves a 3271 PE.

It has been seen in field test sites that the host system must be properly generated to support the RSX-11/3271 PE efficiently. Care should be taken at the host side that appropriate parameters are used in the 3270 system software, parameters that allow reasonable error recovery, appropriate error and timeout thresholds, and half-duplex modes for remote systems.

The following points are also important:

- If a half-duplex connection is to be used, the line must be generated for half-duplex operation. This is necessary because half-duplex modems must have switched RTS/CTS control signals.

- The host system must have a reasonable no-response timeout value for unanswered polls or messages. A typical value is 3 seconds.

- The host system should be reasonably tolerant of protocol errors on the line before dropping a controller from the poll list. The IBM BTAM default is 7 errors in 15 minutes. Bell Telephone only guarantees better than 15 errors in 15 minutes.

- The host system retry count should be at least as high as that of the 3271 PE's. The 3271 PEs default is 7.

Chapter 2
PE Structure and Components

This chapter describes the architecture of the 3271 PE, its components, their relationships, their uses, and the PE generation options that apply to the components.

The 3271 PE system consists of modules that must be added to the RSX operating system to enable it to support the PE communications functions. The modules are the Communications Executive (CEX) and its associated processes and tasks. You will load these modules on top of the RSX operating system after you have completed the PE generation procedure. Chapter 4 describes the generation procedure. Chapter 5 describes the procedure for loading these modules and bringing up a 3271 PE system.

A process is a PE component that runs under the control of the Communications Executive. The following processes are supported by the PE:

- Terminal Unit Driver (TUD)
- Resource Allocation Failure Module (AUX)
- Direct Line Access Process (DLX)
- 3271 Protocol Emulation Module (BSD)
- A driver providing software support for DU11, DUV11, and DUP11 line interfaces (BU)
- A driver providing software support for DV line interfaces (DV)
- A driver for the KMC11-A microprocessor supporting DUP11 line interfaces (KBU)

Each process provides specific functions. TUD, for example, is a pseudodevice driver providing the logical link between application programs and data messages. BSD provides the 3271 protocol emulation.

The RSX Executive is aware of PE processes as read-only common partitions or as dynamic subpartitions in system memory.

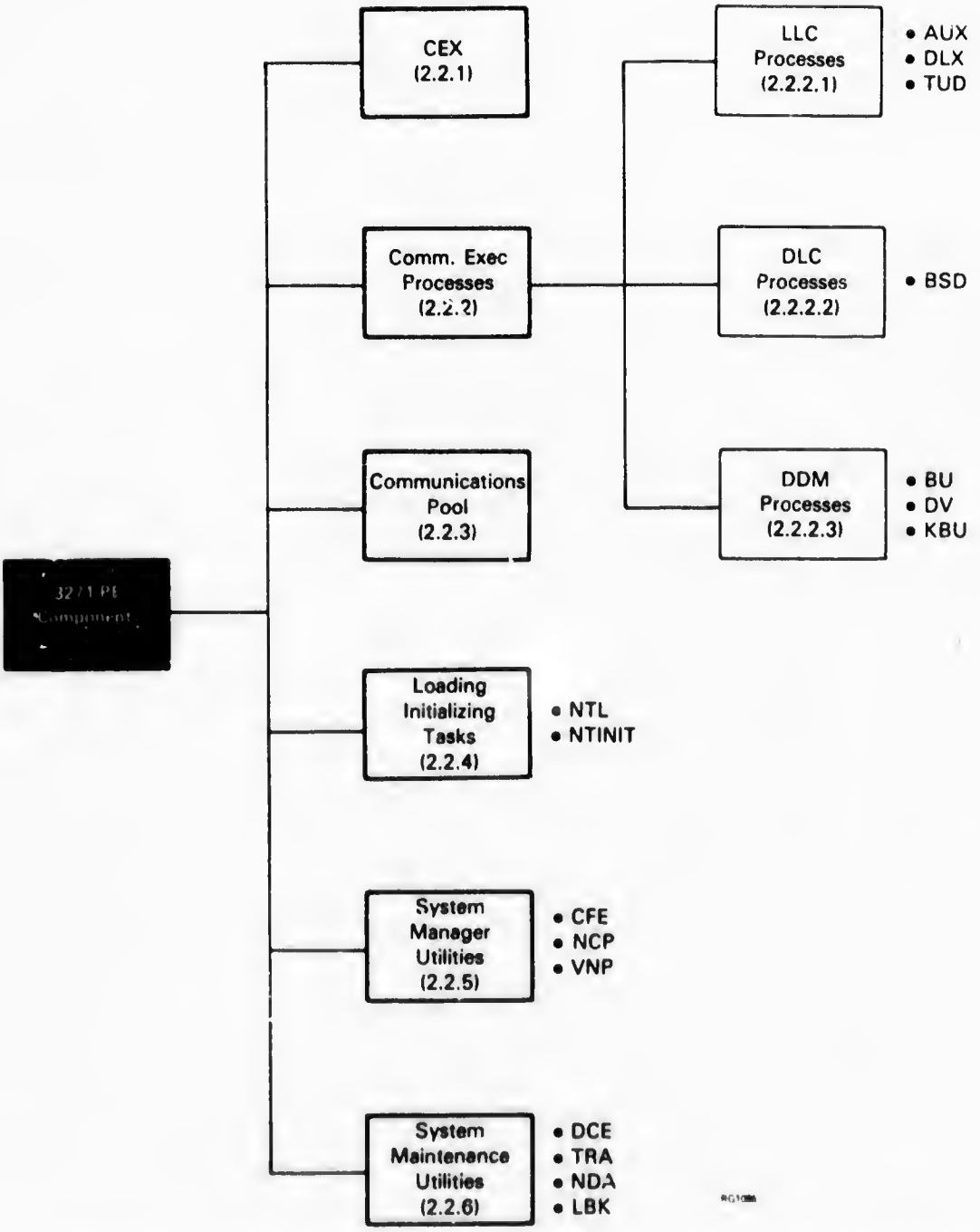


Figure 2-1: Logical Subdivision of RSX-11/3271 PE System Components

The 3271 PE has a number of utilities which run directly under the RSX Executive as program tasks. The PE supports the following tasks:

- Network Loader (NTL)
- Network Initializer (NTINIT)
- Network Control Program (NCP)
- Virtual Network Program (VNP)
- Configuration File Editor (CFE)
- Loopback Test Program (LBK)
- Error Logging and Statistics Routine (DCE)
- Trace Program (TRA)
- Network Dump Analyzer (NDA)

These tasks provide tools for managing and maintaining the system. For example, NCP can be used to load, unload, monitor, and change component parameters. TRA can be used as a debugging tool.

Figure 2-1 illustrates the logical subdivision of the 3271 PE system components. Table 2-1 identifies PE components and related component subgroups, and notes the parts of this manual in which each component is described. Chapter 4 explains how you can select components to be generated as part of your PE system. Table 4-3 shows PE components categorized as processes and tasks.

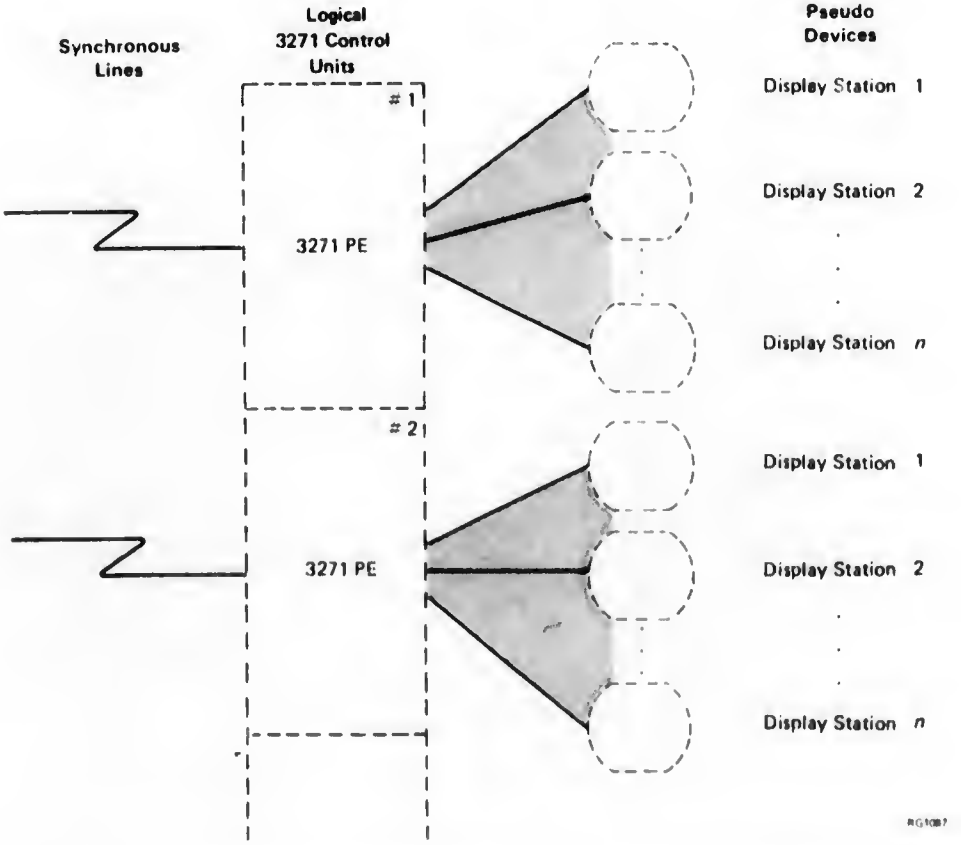
Table 2-1: Component Name/Component Subgroup Cross-Reference List

Component Name	Component Subgroup
AUX	CEX Processes (LLC Processes, 2.2.2.1)
BSD	CEX Processes (DLC Processes, 2.2.2.2)
CEX	CEX (2.2.1)
CFE	System Manager Utilities (2.2.5, 6.6.3.2)
DCE	System Maintenance Utilities (2.2.6, 6.4.2.2, 8.2.2)
DDM	CEX Processes (DDM Processes, 2.2.2.3) Includes BU, DV, and KBU device drivers for PE supported line interfaces.
DLX	CEX Processes (LLC Processes, 2.2.2.1)
LBK	System Maintenance Utilities (2.2.6, 5.2, 8.2.4)
NCP	System Manager Utilities (2.2.5, Chapter 6)
NDA	System Manager Utilities (2.2.6, 8.2.5)
NTINIT	Loading Initializing Tasks (2.2.4)
NTL	Loading Initializing Tasks (2.2.4)
TRA	System Maintenance Utilities (2.2.6, 8.2.3)
TUD	CEX Processes (LLC Processes, 2.2.2.1)
VNP	System Manager Utilities (2.2.5)

2.1 Overview of 3271 PE Architecture

The RSX-11/3271 PE is implemented as a multiple device driver which can control up to six synchronous communications lines and, therefore, emulate six separate IBM 3271 controller units. Each synchronous line can handle a maximum of 32 logical channels (TU pseudodevice drivers). The TU drivers correspond to the display stations on an IBM 3271. Figure 2-2 shows the schema for the 3271 PE lines and pseudodevices.

In the following sections, descriptions of the 3271 PE components will clarify component relationships and how the 3271 PE system functions as a unit. Figure 2-3 illustrates the relationships among the 3271 PE components.



Note: The maximum number of lines and pseudo devices that the 3271 PE can support depends on the system's power and configuration.

Figure 2-2: 3271 PE Lines and Pseudodevices

2.2 3271 PE Components

The 3271 PE components fall into two classes:

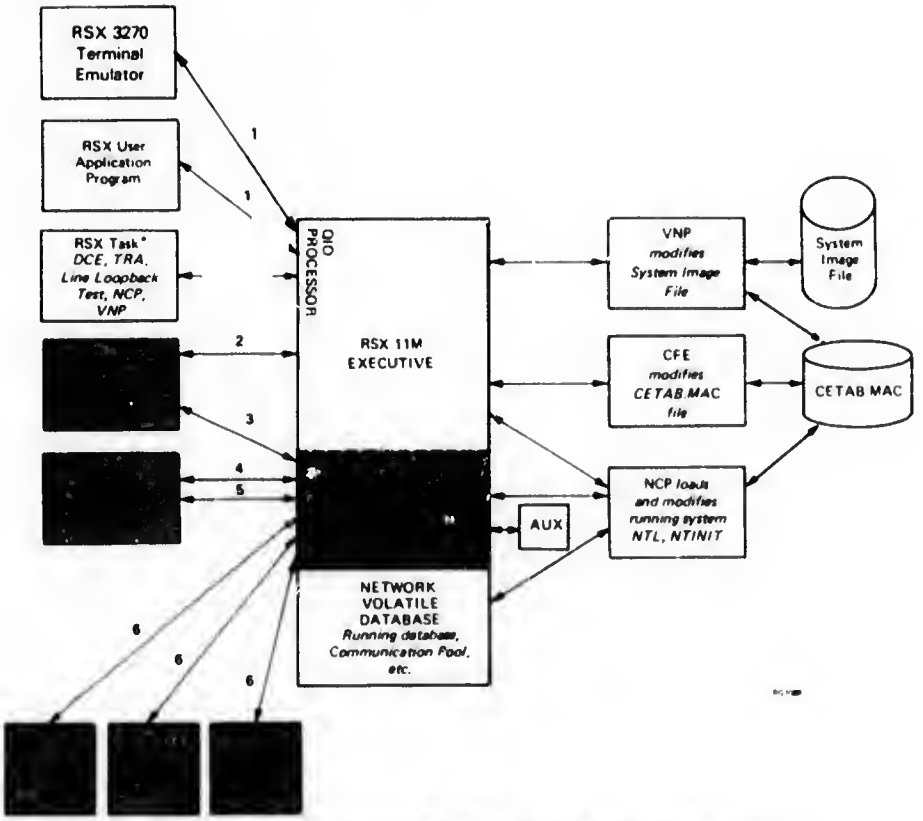
- Those which are resident during the execution of the protocol emulator
- Those which are nonresident and perform services for resident components

The resident components consist of the Communications Executive (CEX), its associated processes, and the Communications Pool (POOL..). Typically these components are loaded by NTL into subpartitions within a system controlled partition. However, you may specifically allocate common partitions for each process. The resident components are:

- Communications Executive (CEX)
- RSX-11M/M-PLUS Pseudodevice Driver module (TUD)
- 3271 Protocol module (BSD)
- Device drivers for the DU11, DUP11, or DUV11 (BU); the DV11 (DV); and the KMC-11A/DUP11 combination (KBU)
- Resource Allocation Failure module (AUX)
- Communication Pool (POOL..)
- Direct Line Access Process (DLX) on RSX-11M-PLUS systems only

The nonresident components of the 3271 PE consist of the loading-initializing tasks, the system management utilities, and the system maintenance utilities. Since these components are not resident, they may share partitions with other nonresident components. The nonresident components are:

- Network Loader (NTL)
- Network Initializer (NTINIT)
- Network Control Program (NCP) and Virtual Network Program (VNP)
- Error Logging and Statistics Routine (DCE)
- Trace Program (TRA)
- Loopback Test Program (LBK)
- Configuration File Editor (CFE)
- Network Dump Analyzer (NDA)



*As monitors, the NCP, VNP, DCE and TRA utilities receive their data from BSD, via TUD, and the QIO processor.

All data transmitted by the RSX-11/3271 PE is first processed at the QIO interface. The RSX 3270 TE and the user application program issue a transmit request (1) which is validated by the RSX Executive QIO processor. The TUD module is notified of the request (2), interprets the request, and facilitates the flow of data to the other PE modules, under the direction of the Communications Executive (CEX) module (3,4,5,6). When the BSD module receives the data (4), it queues the data to the appropriate pseudo device and translates the data from ASCII to EBCDIC. Then, when the BSD module receives a general or specific poll from the host, it adds the necessary protocol characters and queues the data to the appropriate device driver (5,6) for transmission to the host. Incoming data takes the reverse path.

Major components of the 3271 PE system are shaded. The NTINIT and NTL components are invoked by NCP. They read from CETAB.MAC and communicate with the RSX Executive CEX, and the Network Volatile Data Base.

Figure 2-3: Relationships Between 3271 PE Components

The following sections identify the PE components, beginning with the resident components. Generation considerations for each of the components are designated by this symbol ■ .

2.2.1 Communications Executive (CEX)

The Communications Executive (CEX) schedules the processes, manages their interfaces, and provides timer and buffer management services. The Communications Executive resides in RSX Executive address space.

■ *When generating the 3271 PE, your input is requested on the following issues:*

- *For RSX-11M systems, you are asked for the name, base address, and length of the Communications Executive partition.*
- *For RSX-11M-PLUS, you are asked for the value of the symbol $CEX in the RSX-11M-PLUS executive.*
- *If you have DU11s or DUP11s on your system, you can select support for the KG11 communication arithmetic option.*

2.2.2 Communications Executive Processes

There are three classes of Communications Executive processes:

1. Logical Link Control (LLCs)
2. Data Link Control (DLCs)
3. Device Driver Module (DDMs)

These processes communicate with each other by exchanging Communication Control Blocks (CCBs) provided by the Communications Executive. The Communications Executive also uses the CCBs to manage the communications processes and to schedule the destination process for each transfer of information between processes. (CCBs are discussed in Chapter 3.)

2.2.2.1 LLC Processes — A Logical Link Control Process (LLC) handles the communications protocol operating on the logical level. The *channels* that LLCs handle are typically virtual communication paths (that is, the links that exist between the RSX and host application programs). An LLC communicates with a DLC on one side and with another LLC or a task on the other side. It executes as a result of a schedule request from a DLC or another LLC, or as the result of a QIO from a RSX application program.

The 3271 PE LLCs are: TUD, AUX, and DLX.

■ *When generating the 3271 PE, your input is requested on the following issues:*

- *The number of 3271 PE lines on your system*
- *The parameters of the partition containing the process*
- *Maximum transmission block size for the PE (asked for TUD)*

3271 PE Pseudodevice Driver (TUD). The 3271 PE Pseudodevice Driver — or Terminal Unit Driver (TUD) — provides the interface between the RSX application programs and Communication Control Blocks (CCBs). In other words, TUD links RSX application program QIOs to CCBs, and moves the data on through the PE system.

Resource Allocation Failure Module (AUX). The Resource Allocation Failure Module (AUX) performs the functions of buffer recovery, powerfail recovery, modem control (if selected), timer service, and process dispatch. The Communications Executive schedules AUX when resource allocation failures occur.

AUX contains the Communications Executive routines that need not reside in the Communications Executive address space (such as timer support and modem control).

Direct Line Access Process (DLX). The Direct Line Access Process (DLX) provides a user interface to lines at the data link protocol level (PE lines). DLX is always included on RSX-11M-PLUS systems. DLX is used for turning lines on and off, and for reconfiguring with the 'CON' task in RSX-11M-PLUS systems (see Section 6.6.5)

2.2.2.2 DLC Processes — Data Link Control processes (DLCs) handle the physical link (frame level) protocol for all the communications lines on the system that use the Binary Synchronous protocol. A DLC process communicates with a DDM on one side and with an LLC on the other. A DLC executes as a result of a schedule request from a LLC (to transmit data) or from a DDM (to receive data from the host).

There is only one DLC process in the 3271 PE system: the 3271 Protocol Emulation module (BSD).

IBM 3271 Protocol Emulation Module (BSD). The IBM 3271 Protocol Emulation (BSD) module provides the IBM protocol emulation. It receives user transmit and receive buffers from the TUD component, and queues them internally until the host polls or selects that 3271 control unit which the PE is emulating. It performs the necessary translation and passes the data to a Device Driver Module.

The BSD translates received data in the user buffer from EBCDIC to ASCII and gives the RSX user application program a completion signal. BSD translates data that is to be transmitted to the host from ASCII to EBCDIC.

In addition, BSD maintains statistics on the number of messages sent and received, and on the number of errors encountered. These statistics are printed by the DCE and the NCP components.

■ *When generating the 3271 PE, you must provide partition parameters for BSD.*

NOTE

The 3271 PE process, BSD, has five timers that can affect the operation of the PE under unusual conditions. These timers are used to timeout certain operations. They are found in the UIC of your target system in the file [(group),10]BSDX.MAC. If you have timeout problems, you can modify these timers by changing the value in BSDX.MAC. Then reassemble BSDX and relink BSD using BSDBLD.CMD.

2.2.2.3 DDM Processes – A Device Driver module (DDM) is supplied for each line interface in the PE system and controls the physical device in much the same way that a disk driver controls a disk for the RSX operating system. The *channels* that DDMs handle are the communications lines controlled by the physical device. (Proper identification of lines is discussed below.) A DDM communicates with a device on one side and a DLC on the other. It executes as a result of either a device interrupt or a schedule request from a DLC.

The DDMs supported by the 3271 PE system are BUs, KBUs, and DVs.

■ *When generating the 3271 PE, your input is requested on the following issues:*

- *The partition parameters for each DDM.*
- *The number of each device driver (controller) type and their hardware priorities, CSRs, and vectors. For example, if your hardware configuration includes one DU11 and one DUP11, then you will need two BUs in your system.*
- *The emulated controller address for each line, as used by the host system to poll the 3271 controller.*
- *The station address for each station on a line. The address is to be used by the host system to poll the individual display stations.*
- *Additional device-specific information about each communications line on each controller (line interface).*

Each of the device drivers supported by the 3271 PE is described below.

BU Driver. The BU driver provides support for the DU11, DUV11, and DUP11 single-line synchronous interfaces. The BU can determine which device it is interfacing with and treat it appropriately.

DV Driver. The DV provides support for the DV11 synchronous multiplexer. The DV hardware supports DMA (Direct Memory Access) transfers and CRC calculations for multiple lines. (Direct Memory Access means that data transfers are made directly to and from memory without intermediate storage in a register.)

KBU Drivers. The KBU driver provides support for a KMC11-A microprocessor on DUP11 synchronous line interfaces. The primary reason to use the KBU is to offload the character processing overhead of running the DUPs. The KBU microcode greatly reduces processor overhead during idle time on the line. The KMC processor will respond to idle polls from the host system without interrupting the PDP-11 CPU as long as the PE has indicated that it has nothing to send. The KBU also reduces overhead during reception and transmission by only interrupting on a per buffer basis. The KMC11-A also does DMA transfers and CRC checking for the 3271 PE.

Line Identification. The system manager can control and monitor many aspects of line activity (Section 6.2.3). Commands affecting lines require line identification (in use of NCP, VNP or CFE, for example). To identify lines, enter one or more keywords followed by an identification string, if necessary. There are three basic formats for line identification:

Keyword	Variable
LINE	*line-identification*
ACTIVE LINES (NCP/VNP only)	(no variable)
KNOWN LINES	(no variable)

KNOWN LINES are all lines identified to the system during PE generation. ACTIVE LINES are all known lines that are currently turned on.

You identify a particular line using the keyword LINE and the unique identification of that line that is in the configuration file, CETAB.MAC, created during generation of the PE. If you want to create another line identification in the configuration file, you have to do another generation.

Line identification has the following format:

dev-c-u

where:

dev is a device driver name (BU, KBU or DV).

c is a decimal number (0 or a positive integer) designating the device driver's hardware controller.

u is a decimal unit/line number (0 or a positive integer) included if the device is a multiple line controller.

Since the BU driver is a single-line device, the format for BU lines is dev-c.

An asterisk (*) signifies a wild card that replaces a controller, or a unit number. The asterisk specifies all known lines in the range indicated by its position in the line identification. You can use more than one wild card in a line identification, but a wild card cannot precede a controller or unit number.

Examples of legal line identifications:

Line Identification	Meaning
BU-*	All known BU controllers
DV-3-*	DV, controller 3, all known units
KBU-3-4	KBU, controller 3, unit 4
BU-0	BU, controller 0

Examples of illegal line identifications:

Do not use * to mean all devices.

Do not use DV-*-3 to mean all unit 3's on all controllers.

Do not use BU-0-1; BU is not a multiple line controller.

2.2.3 Communication Pool (POOL..)

The Communication Pool (POOL..) is a common area where many of the common subroutines and data tables reside. The receive data buffers are taken from this area. The Communication Pool contains the mode tables used by the DDMs. (Mode tables are used by the DV11 to determine how to handle any character received or transmitted.) Also, the Cyclic Redundancy Check (CRC) calculation routine is loaded here by NTL. This routine, known as STCRC to the PE, is tailored during PE generation for either the KG11 or CRC software.

The remaining components discussed are non-resident.

2.2.4 Loading Initializing Tasks

Loading initializing tasks are the Network Loader task (NTL) and the Network Initializer task (NTINIT).

- *When generating the 3271 PE, your input is requested on the following issues:*
  - *Partition parameters for NTL and NTINIT*
  - *Whether your configuration has a KBU(KMC11-A) device driver*

Network Loader (NTL). The Network Loader (NTL) is a subsidiary task invoked by NCP. NTL loads and unloads the 3271 PE system software components in a running RSX-11M or M-PLUS system. NTL allocates the necessary line tables from the executive pool and sets up associated data structures. NTL also performs the clear function; that is, NTL removes the 3271 components and associated data structures from the RSX system, returning the used resources to the RSX system.

Network Initializer (NTINIT). The Network Initializer (NTINIT) is invoked by certain NCP, NTL, or VNP requests. It may also be automatically invoked during powerfail recovery. (NTINIT restarts timers and loads KMC devices.) NTINIT loads the Communications Executive into memory and allocates the CCBs. On RSX-11M systems, the CCBs are allocated first from any available space in the Communications Executive partition, then from system dynamic memory. On RSX-11M-PLUS systems, CCBs are allocated from the Dynamic Storage Region (DSR). Then, NTINIT allocates the POOL.. partition if it does not already exist. It allocates the receive data buffers from the communications pool partition, and completes a marked-for-load scan, generating load requests to NTL for components marked for load with the Communications Executive. Note that NTINIT is required if you intend to load KBU (KMC11-A) microcode with VNP.

2.2.5 System Manager Utilities and the PE System Data Bases

The system manager utilities consists of NCP, VNP, and CFE. These utilities affect different data bases. It is therefore necessary to understand the differences between the data bases which will now be described:

2.2.5.1 PE System Data Bases –

Running data base. This is the PE system image currently in memory. When you first bring the system up, the running data base includes information from the configuration file CETAB.MAC. You can change this running data base with NCP commands. The changes will stay in effect until the system is unloaded.

Permanent data base. The permanent data base is created in the configuration file, CETAB.MAC, during PE generation. CETAB.MAC contains the values for all the components and parameters of your system. The CFE utility modifies the CETAB.MAC permanent data base. Changes take effect only when the system is reloaded.

System image file. The system image file is the data base that contains a binary image of your RSX-11 operating system. The VNP utility can be used to modify the system image file.

2.2.5.2 System Management Utilities –

Network Control Program (NCP). The Network Control Program provides the following services:

1. It sets, changes, and clears PE system component states and parameters (Sections 6.2.1, 6.2.2, and 6.2.3).

2. It displays network counters and parameters (Section 6.5.1).

NCP commands affect the data base of the running system. If the system goes down, or is rebooted, any changes made using NCP are lost. A system is booted from the disk system image file. The disk system image file may contain PE software if the software was put into that file using VNP. Otherwise, the PE software must be loaded into memory using NCP.

The subset of NCP commands supported by the 3271 PE is presented in Appendix E.

■ *When generating the 3271 PE, your input is requested on the following issue:*

- *Partition parameters for NCP.*

Virtual Network Program (VNP). The Virtual Network Program is optional on RSX-11M systems and not available on RSX-11M-PLUS systems. VNP may be used to modify an RSX-11M disk system image in a fashion similar to the use of the Virtual Monitor Routine (VMR) on the RSX operating system image. The VNP performs the same functions as NCP, except it will not change the PE software configuration in main memory. With VNP, the changes made to the system image file apply only to the software the next time the operating system is booted. However, the changes are permanent in the image file until explicitly reset. Note that if you intend to use VNP to load a system with a KBU (KMC-11A) microcode, you must have NTINIT installed in your system.

Before using VNP, it is necessary to have the system image file in the UIC in which the PE software task images reside. The system image can be copied using Peripheral Interchange Program (PIP). If the contiguous space on your system disk is insufficient to copy the system image, first copy it to a backup medium, then rename the system image to the PE software UIC by using the PIP /RE switch. Always back up your system image before using VNP.

The VNP commands that you can use with the 3271 PE are described in Appendix F.

■ *When generating the 3271 PE, your input is requested on the following issue:*

- *Partition parameters for VNP.*

Configuration File Editor (CFE). The Configuration File Editor modifies parameters that are defined in the configuration file (CETAB.MAC) created during generation of the PE. CFE commands allow:

1. Setting, changing, and clearing line, process, and system parameters

2. Displaying line, process, and system parameters

The CFE commands that you can use with the 3271 PE are fully described in Section 6.6.3.2 and Appendix D.

■ *When generating the 3271 PE, your input is requested on the following issue:*

- *Partition parameters for CFE.*

2.2.6 System Maintenance Utilities

The system maintenance utilities are:

- Error Logging and Statistics Routine (DCE)
- Trace Program (TRA)
- Line Loopback Test Program (LBK)
- Network Dump Analyzer (NDA)

Error Logging and Statistics Routine (DCE). The Error Logging and Statistics Routine provides formatted reports of statistics accumulated by the BSD process. DCE reads and resets the statistics for each active line of the PE system. After formatting the statistics, DCE places the reports in a log file for use by the system manager. DCE is a nonprivileged applications task. DCE can be used as a line monitor (Section 6.4.2.2) or as a debugging tool (Section 8.2.2).

■ *When generating the 3271 PE, your input is requested on the following issues:*

- *Partition parameters for DCE.*
- *Device, UIC, and file name for the log file.*

Trace Program (TRA). Trace enables you to see how the BSD module is exercising the protocol relative to specific lines. TRA lets you have a realtime view of messages exactly as they are transmitted and received by the 3271 PE. It also retrieves messages and associated codes used in controlling communications between the BSD module and the DDMs. The program is an optional PE component which you can select during PEGEN. It is a nonprivileged task. TRA is most useful for program and line debugging. (See Section 8.2.3 for details on operating TRA.)

- *When generating the 3271 PE, your input is requested on the following issue:*

  - *Partition parameters for TRA.*

Line Loopback Test Program (LBK). The Line Loopback Test Program is a hardware diagnostic utility that allows you to test communications hardware and monitor communications lines. The LBK monitoring facility provides a complete trace of any data coming in on a line, whereas the Trace Program provides a trace on a per station basis. Note that while in loopback mode the PE does not respond to normal polling on the line. See Section 8.2.4 for details on operating LBK.

LBK tests the software configuration and the hardware by looping back a test message through a manually set hardware device.

LBK requires that you have the Trace feature generated with your PE.

- *When generating the 3271 PE, your input is requested on the following issue:*

  - *Partition parameters for LBK.*

Network Dump Analyzer (NDA). The Network Dump Analyzer is a system maintenance utility that is optional with the 3271 PE. NDA can operate on crash dumps produced by the RSX-11M/M-PLUS Executive Crash Dump option (this means that you have built the Executive Crash Dump routine during system generation). NDA produces a formatted listing of the 3271 PE system data structures in the crashed system. NDA is only useful if you have a knowledge of the PE internals and data bases, or for reporting problems. Operation of NDA is explained in Section 8.2.5.

- *When generating the 3271 PE, your input is requested on the following issue:*

  - *Partition parameters for NDA.*

Chapter 3
Pre-Generation Requirements and Considerations

This chapter discusses considerations to be exercised prior to generating your 3271 Protocol Emulator. It specifies system characteristics required for 3271 PE support and tells how to revise your system.

The 3271 Protocol Emulator Generation procedure (PEGEN) allows you

- To tailor the PE to your needs
- To task-build the components and data structures
- To integrate them into your RSX operating system

PEGEN is similar to the generation procedures for RSX (SYSGEN) in that it involves executing an indirect command file and answering questions displayed on your screen.

You must perform the PEGEN generation procedure to add the 3271 PE to your system. If you want to make a major change to the 3271 PE after generation, you must regenerate the product. Major changes include:

- Adding processes
- Changing the number of communication lines
- Changing the number of TU devices (stations)
- Changing communications controllers (for example, changing from a BU to a KBU)

You can make other changes to the systems without requiring regeneration by using CFE, NCP, or VNP. See Section 6.6.3.

Before generating your 3271 PE, you must have a clear idea of the environment in which it will operate and of the functions you will want to perform. You must know the number and types of devices, lines, and components in your system. Also, your RSX operating system must have certain features required by the PE. This may mean regenerating your system by means of SYSGEN.

This chapter covers the following issues:

1. Specifications for RSX-11M or M-PLUS SYSGEN parameters that influence the RSX-11/3271 PE

2. Requisites for generation of the 3271 PE

3.1 RSX-11M/M-PLUS Considerations

To generate a RSX system supporting the 3271 PE, you must supply the appropriate answers to several SYSGEN questions. The relevant SYSGEN parameters, both mandatory and optional, are discussed in the following sections.

3.1.1 Customizing the RSX Executive for PE Support

To support the 3271 PE, certain special support features must be included in the RSX Executive. A YES response to the following question in the Executive Options section of the SYSGEN dialog will give you these features:

For RSX-11M systems:

```
>* 32. Include support for communications products? [Y/N]:
```

For RSX-11M-PLUS Systems:

```
>* Do you want support for communications products (such as DECnet)?
>; [Y/N D:N]:
```

Further, you must create a common partition for the PE Communication Executive within the RSX-11M Executive address space. (In RSX-11M-PLUS systems the Executive automatically takes care of this.) Any space which is not used for this partition will be used for the Communications Control Blocks (CCBs) pool, which must come from system dynamic memory. Therefore, you should make the Communications Executive as small as possible.

You create the Communications Executive partition, called CEXPAR, during RSX-11M SYSGEN at the stage at which you use the Virtual Monitor Routine (VMR) on the new system image. Perform the following two steps:

1. **Set the system pool** down from the top of the RSX Executive, leaving only space enough for the PE Communication Executive partition, CEXPAR, as shown in Figure 3-1.

   For example, if you have a 16K (100000 octal) Executive and you want CEXPAR to be 5000 octal bytes long, subtract 5000 from the top address of the RSX Executive to find the top of the pool. Thus, given that the top of your Executive is 100000 octal, subtracting 5000 octal from this you obtain 73000 octal, the top of the pool. Using VMR, the following command sets the top of the pool at 73000 octal:

   ```
   VMR)SET /POOL=730
   ```

   Note that VMR allocates space in 100 octal blocks. Thus, 730 is used as the top of the pool, instead of 73000.

2. **Define the Communication Executive partition** as a main common partition set from the top of the RSX Executive with its base address at the top of the system pool.

   Continuing with the example given in step 1 above, the following VMR command defines CEXPAR as a main common partition with a base address of 73000 octal, and a length of 5000 octal bytes:

   ```
   VMR)SET /MAIN=CEXPAR:730:50:COM
   ```

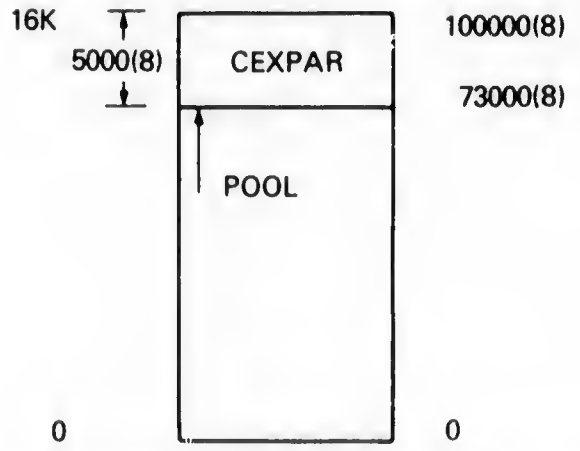


Figure 3-1: Allocating the Communications Executive Partition

3.1.2 Identifying Communications Devices

Because the RSX-11M or M-PLUS Executive does not need to know anything about hardware that will be used as PE communications devices, do not specify any of those devices when answering questions in the Target Configuration section of SYSGEN. However, when you generate your RSX system, you must specify how much space is to be allocated for interrupt vectors on your target system, and the space must be sufficient to include the interrupt vectors belonging to the communication devices. Answer the following question with the highest vector address of all devices on your system, including the communication hardware.

For RSX-11M systems:

```
>*14. Highest interrupt vector [0 R:0-774 D:0]:
```

For RSX-11M-PLUS systems:

```
>* What is the highest interrupt vector address? [0 R:n-774 D:n]:
```

where

n is the highest vector address of devices specified in SYSGEN

(RSX SYSGEN automatically sizes the system stack according to the highest of the interrupt vector addresses. The higher you set the interrupt vector address, the larger will be the system stack created by SYSGEN. If the vector address is set too low, the stack might overflow due to excessive interrupts.)

3.1.3 Treatment of Parallel Questions in PEGEN and SYSGEN

On certain topics, parallel questions are issued in both SYSGEN and PEGEN displays. Your responses to this set of questions must be the same for both SYSGEN and PEGEN. Responses depend upon the processor and software used in your configuration. (Note: Some of these questions are never asked during SYSGEN of a RSX-11M-PLUS system.)

The following SYSGEN questions have equivalent PEGEN questions:

Target Configuration section of SYSGEN

```
>*1. Processor Type [D:11/34][S]:
```

```
>*3. Memory size (in K-word blocks) [D R:16.-124. D:16.]:
```

Setup questions

```
>*5. Are you generating a mapped system? [Y/N]:
```

Note that you must have a mapped RSX system to support the 3271 PE configuration. If memory size is greater than 28K you will automatically have a mapped system.

Executive Options section of SYSGEN

```
>* 7. Memory management (PLAS) directives? [Y/N]:

>*12. Multi-user protection? [Y/N]:

>*20. Power fail recovery? [Y/N]:

>*33. System controlled partitions? [Y/N]:

>*49. Checkpointing: A-Yes B-With system ckpt. file [S]:
```

3.1.4 Loading Options

Optional SYSGEN parameters allow you to:

- Load all network tasks into one partition and all network processes into dynamically created subpartitions. To do this, select system controlled partitions (Question 33 in the Executive Options section of SYSGEN).

  See Section 3.3 for detail on partition layout.

- Include crash dump support handled by the NDA utility. To do this, you must first build the Executive Crash Dump Routine into your system during phase 1 of an RSX-11 SYSGEN. This is accomplished by entering B in response to the query:

  ```
  *50. Dump Analysis: A- PANIC B- Crash Dump Analysis (CDA)
  ```

  If you select this option, code will be included in your system to dump memory to a secondary storage device following a system crash. Your secondary storage device can be any one of a number of different types of disk units, a magtape unit or a DECtape unit.

3.2 Generation Requirements

Before generating your Emulator, establish the following:

- The number of terminal stations supported
- The number and types of communications devices and lines supported
- The addresses of the control unit and devices to be used by the host system
- The number of communications lines that will be in use at a given time
- The number of transmits and receives that may be outstanding at a given time
- The size of the message blocks (LDBs) sent by the host system

- The number of Communications Control Blocks (CCBs)
- The number of Receive Data Buffers (Large Data Buffers)
- The device priorities
- The device CSR and vector addresses
- The partition names

You must also make sure that your system has enough memory to run the independent utilities and components to be used by the PE and that enough memory is available for the network pool. Finally, for RSX-11M operating systems, a partition must be set aside for the Communications Executive. The latter conditions can be taken care of through SYSGEN. See above for details on SYSGEN and further pregeneration considerations. Section 3.3 of this chapter provides suggestions for partition layout; Appendix B provides approximate partition sizes. Section 3.4 discusses memory allocation requirements for buffers.

3.3 Partition Layout

All PE software and network processes require common partitions. If they are dynamically loaded into a system controlled partition, they will have subpartition names of NT.*xxx* where *xxx* is the process name.

Since these common partitions cannot be shuffled or checkpointed, you must be careful not to fragment a system-controlled partition that must be assigned to other tasks, network software, and the PE.

There are three ways to avoid fragmentation:

1. Having a separate system-controlled partition for the processes.
2. Specifying top-down loading in PEGEN. This causes all processes (that are to be loaded at start-up time) and the network pool to be loaded from the top of a system-controlled partition. You should also use the TOP argument in any NCP SET LINES command.
3. Using VNP to load the processes in the disk image file of the system (RSX11M.SYS).

If you build the Network Loader (NTL), the Network Initializer (NTINIT), and the Network Control Program (NCP) for the same system controlled partition as the PE software or network processes, you must observe the following guidelines for start-up to succeed:

1. NTL and NTINIT are slave tasks of NCP. Therefore, the system controlled partition must be larger than the combined sizes of NTL, NTINIT, NCP and any processes and the network pool that are loaded at start-up.

2. If the PE is to be started from an indirect command file and the indirect command file processor (AT.) is also installed in the same system-controlled partition as NTL, NTINIT, NCP and the processes, then the size of AT. must also be included in the above guideline.

The following sequence of events illustrates the restriction:

1. The PE is to be started from an indirect command file, so AT. invokes NCP.

2. NCP runs and invokes the initializer, NTINIT.

3. NTINIT runs and invokes the loader, NTL.

Assuming that the partition was at least as large as the sum of the sizes of AT., NCP, NTINIT, and NTL, they are all resident in the system-controlled partition. Keep in mind that NTL cannot checkpoint a task in order to load a process. Therefore, there must be sufficient space left in the partition to load each process that is specified by the NCP command and to load the network pool.

3.4 Buffer Considerations

The 3271 PE uses Large Data Buffers (LDBs) and Communications Control Blocks (CCBs) to perform communications functions. During PEGEN you must determine the amount of physical memory allocated for each type of buffer. (Buffer parameters can be changed later using CFE.) Functional descriptions of the buffers and some considerations for determining buffer parameters in your PEGEN dialog follow.

Large Data Buffers. The 3271 PE uses LDBs as receive buffers for intermediate storage of all incoming messages. With Unibus Mapping Registers (UMR) mapped systems using a KBU or DV device driver, LDBs are also used as transmit buffers for intermediate storage of user data being sent. (In all other cases, user data buffers are used as transmit buffers.) During PEGEN, you will specify both the size and number of LDBs that are to be allocated. Once a size is set, it remains fixed, unless modified by CFE. The modified size will then become effective the next time the PE is loaded.

You must also specify the minimum number of LDBs that are reserved for use as receive buffers. This enables you to give receive traffic priority over transmit traffic. When the number of available LDBs is equal to or less than the specified minimum, only requests for receive buffers will be serviced. (Requests for transmit buffers will be queued until the number of available LDBs is above the minimum.)

Communications Control Blocks. The Communications Executive uses CCBs to schedule the PE processes. The size of CCBs is fixed at 28. bytes; you do not determine the size. During PEGEN you specify the number of CCBs to be allocated.

3.4.1 Allocation of Memory for Buffer Space

Deciding how much memory to dedicate to buffer space always involves tradeoffs between minimizing memory usage and optimizing performance. The information provided here is for all PE users who want optimal memory utilization and performance.

Table 3-1 provides the following information about each buffer type:

- How the buffers are used
- Where the buffers are located
- The size of the buffers
- Factors that determine the number of buffers required

Table 3-2 gives the recommended number of LDBs for each communications device supported by the 3271 PE. Both tables provide adequate information for determining buffer resource requirements for your PE system. While planning for the amount of memory to dedicate to buffer space, remember that you can adjust the number of LDBs and CCBs, and the size of LDBs at a later time. Do this by using CFE's DEFINE SYSTEM command, discussed in Section 6.6.3.2.

All PE large buffers are located in the network buffer pool (Communications Pool). The default buffer pool is named POOL.. and is created as a common partition.

All processes must be mapped to CCBs. To facilitate this mapping, CCBs are allocated in executive address space. CCBs are first allocated from the end of the Communications Executive partition if it is not completely filled. If more CCBs are needed and if the available space allows, they are allocated from the Executive Dynamic Storage Region (DSR).

NOTE

If you have a UNIBUS Mapping Registers (UMR) mapped system with DV or KBU device drivers, the PE restricts the transmission segment size to the size of the LDB. This restriction makes possible the correct UMR mapping for the device drivers. See Section 3.4.2 for further explanation.

3.4.2 Large Data Buffer (LDB) Requirements

During PEGEN you specify a LDB size in the range of 192. to 1024. bytes. The default size is 280. bytes. The minimum LDB size for a UMR mapped system with a DV or KBU device driver is equal to the minimum segment size for transmission. The user application program has the option of sending a message as a series of segments. Each segment is framed by BISYNC control characters. If messages are not segmented, then the segment size is equivalent to the transmission block size. For other systems it is possible to use a smaller buffer.

In selecting an LDB size, you must consider the following factors:

- The transmission segment size used by your application programs
- The amount of memory that you wish to allocate to LDBs
- The throughput requirements
- The host system requirements

Choose a size large enough to hold typical messages, yet leave enough pool to contain approximately two LDBs per line.

For a BU device driver, the actual LDB size for user data is two bytes less than specified LDB size (the BU driver reserves these two bytes for the CRC).

Note that for receive operations the 3271 PE can link multiple LDBs for messages longer than the LDB size. Also, note that for each LDB, a CCB is automatically allocated by the COMM EXEC. Therefore, in allocating memory for LDB space, plan for an extra 28. bytes from the Dynamic Storage Region (DSR) or COMM EXEC partition for each LDB allocated. However, when determining the number of CCBs to specify in PEGEN, do not include in the total those CCBs allocated for LDBs.

The last row in Table 3-1 summarizes some of the considerations for determining the number of buffers that are required to maintain a reasonable level of performance. The number of LDBs required, as shown, depends on the device types and the number of controllers of each device type in your configuration. PEGEN allows you to choose from 3 to 200 LDBs. Remember UMR mapped systems with DV or KBU device drivers require one LDB for each user transmit outstanding.

Table 3-2 gives the recommended number of LDBs for each communications device supported by the 3271 PE.

Examples illustrating how to determine the required number of LDBs for a PE system are given following the tables.

Table 3-1: PE Buffers

	Buffer Type	
	Large Data Buffers	**Communications Control Blocks**
Use	All receive messages. User data transmissions.**	Interprocess communication
Location	Network Buffer Pool, common partition (POOL..)	Executive address space
Size (in Decimal)	Variable** Range: 192 to 1024 bytes	28 bytes
Factors Determining Number of Buffers Required*	Number of lines. (See Table 3-2) Device types.	Number of lines simultaneously starting; number of user application program transmits outstanding. 3 per active PE line, 2 extra per line, only while starting

Notes:

* The required number of buffers assumes the base level situation. When lines are active simultaneously, the number increases. When a line is off, the PE does not require LDBs or CCBs for that line. However, if you intend to use that line, it is advisable to allocate the resources needed to run the line.

** Large data buffers are used as transmit buffers only on UMR mapped systems with a DV or KBU. On such systems, the transmit segment size cannot exceed the LDB size. There is no restriction on the receive message size for any PDP-11 system.

Table 3-2: LDBs Required for Devices Used by the PE

Device	Recommended Number of Large Buffers
BU	One for each active line plus the maximum message size divided by the LDB size, and add 1 more LDB if there is a remainder.
DV	Two for each active line plus the maximum message size divided by the LDB size, and add 1 more LDB if there is a remainder. Minimum of 2 always required when ON.
KBU	Two for each active line plus the maximum size divided by the LDB size, and add 1 more LDB if there is a remainder. Minimum of 2 always required when ON.

To determine the minimum number of LDBs

For Example:

Assume you have two BUs, BU-0 and BU-1. Assume also that the largest message size is 2001. bytes and the LDB size is 250. bytes. Then, using Table 3-2, you can determine the number of LDBs required for your system as follows:

Step 1. Determine the number of LBDs required for each BU.

Each BU line takes one plus the maximum message size divided by the LDB size (1 + 2001/250 = 9). Since the division renders a remainder, add one more LDB to the sum.

The number of LBDs required for each BU = 10

Step 2. Determine the number of LBDs required in total.

Double the number obtained in Step 1 to obtain the total number of LDBs required for two BUs.

The total number of LBDs required = 20

3.4.3 Communications Control Block (CCB) Requirements

PEGEN allows you to choose from 4 to 200 CCBs. Three CCBs are required per PE line and two extra CCBs are required for each line when starting (that is, when turned to the ON state). After start up, these latter two are released. The COMM EXEC generally uses an additional CCB for each user request for the duration of the requested operations.

In non-UMR mapped systems, one CCB is automatically allocated for each receive and transmit. In UMR mapped systems, a CCB is allocated for each receive, and an LDB for each transmit. If you queue several receives or transmits, additional CCBs (and/or LDBs) will be required. Be sure to specify in PEGEN a larger number of CCBs, based on the following:

- Number of TU units that you have active
- Number of receives and transmits (non-UMR systems only) outstanding

NOTE

CCBs require dynamic memory. Remember you can adjust the number of CCBs later to tune the system performance (using CFE).

To determine the minimum number of CCBs:

Assume you have the same configuration as used in Standalone example in Section 3.4.2, for determining the number of LDBs; two BUs, BU-0 and BU-1. Assume also that BU-0 will be in ON state and BU-1 will be OFF. Then determine the minimum number of CCBs required for your system as follows:

Step 1. BU-0 (ON)

Determine the number of CCBs for the BU in ON state. Referring to Table 3-1 you can see that a BU line takes three CCBs plus two extra while in starting state. Let 5 be the minimum number of CCBs required for this line.

5

Step 2. BU-1 (OFF)

Determine the number of CCBs for the BU in OFF state. A line in OFF state does not require any CCBs, but if you are considering using this line at a later time, you should allocate at least 3 CCBs for this line (you will need 2 more while starting).

3

Step 3. Total

Now take the totals obtained in Steps 1 and 2 to obtain the minimum number of CCBs required. In this case, the total is eight.

Total 8

3.5 Assignments of Control Unit and Station Addresses

During PEGEN you will be asked for the address of each controller line and of each station on a controller. The addresses that you specify will be the ones that the line will recognize during polling from the host system. PEGEN provides a translation table that helps you select the proper addresses (Table 3-3). An example of the assignment of addresses during PEGEN follows.

Assume your 3271 PE system communicates by means of two BU controller lines as in the configuration shown in Figure 3-2. To the host system your PE appears as two 3271 control units. One controller (BU-0) has two stations or devices attached to it, while the other (BU-1) has three. The addresses of the units and stations as known to the host system are given in the figure.

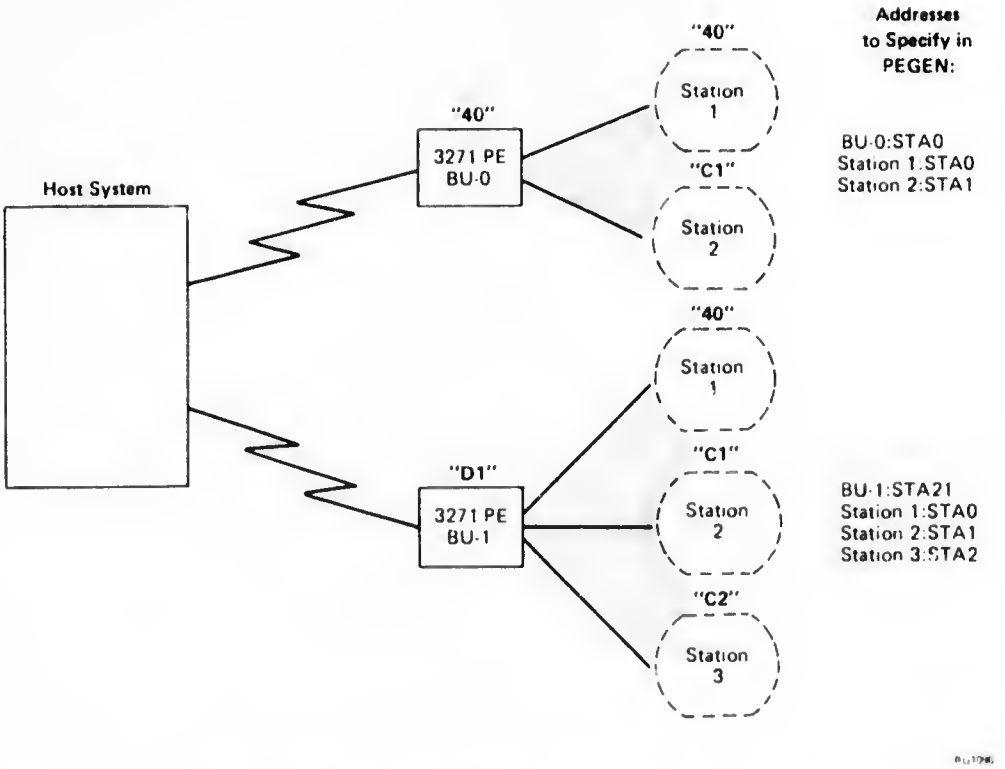


Figure 3-2: A 3271 PE Configuration with Two BU Controller Lines and Their Stations (Pseudodevices)

PEGEN asks you for the number of stations on each of the RSX system's controllers to be polled by the host system. For BU-0 your response should be 2 and for BU-1 your response should be 3.

A subsequent question in PEGEN will ask for the controller address for BU-0. If you press ⟨ESC⟩ in response to this question. PEGEN displays explanatory material and the translation table shown in Table 3-3.

Table 3-3: Controller and Station Address Translation Table

CU or Device Symbol	ERCDIC I/O Char	Hex	Octal	CU Number Octal	CU Number Decimal
STA0	SP	40	100	0	0
STA1	A	C1	301	1	1
STA2	B	C2	302	2	2
STA3	C	C3	303	3	3
STA4	D	C4	304	4	4
STA5	E	C5	305	5	5
STA6	F	C6	306	6	6
STA7	G	C7	307	7	7
STA10	H	C8	310	10	8
STA11	I	C9	311	11	9
STA12	CENT	4A	112	12	10
STA13	.	4B	113	13	11
STA14	(	4C	114	14	12
STA15	(	4D	115	15	13
STA16	+	4E	116	16	14
STA17	BAR	4F	117	17	15
STA20	&	50	120	20	16
STA21	J	D1	321	21	17
STA22	K	D2	322	22	18
STA23	L	D3	323	23	19
STA24	M	D4	324	24	20
STA25	N	D5	325	25	21
STA26	0	D6	326	26	22
STA27	P	D7	327	27	23
STA30	Q	D8	330	30	24
STA31	R	D9	331	31	25
STA32	!	5A	132	32	26
STA33	$	5B	133	33	27
STA34	*	5C	134	34	28
STA35	)	5D	135	35	29
STA36	;	5E	136	36	30
STA37	'	5F	137	37	31

Now you must choose a controller address to match the emulated controller address used by the host for polling. (Chapter 7 shows the format of the host polling message that includes the polling address.) The second column of Table 3-3 lists the one-character polling addresses of 3271 control units, as known to the host. This address may be coded by the host system with the corresponding hexadecimal representation shown in the third column of the table. According to Figure 3-2 the addresses are known to the host system as hexadecimal representations. For instance, the address of controller BU-0 as known to the host system is hexadecimal 40. The matching address (symbol) that you must specify in PEGEN is found in the first column of the table directly across from hexadecimal 40. Thus, during PEGEN you would specify STA0 as the controller address for BU-0.

Note that the host system can give a controller on a line a unique address representation, or it may give a representation that is shared by other controllers if they are each located on physically distinct lines. For example, both BU controllers above could be coded with the address of hexadecimal 40.

Next, PEGEN asks for the address of each station on controller line BU-0. Answer this as for above: the station address (symbol) must match the corresponding station address known to the host system. Typically (but not necessarily) the host assigns consecutive addresses begining with hex 40. As shown in Figure 3-2, the addresses of the stations on BU-0 are 40 and C1. Use the same translation table to determine the proper match for controller stations. You should give station 1 the address of STA0, and station 2 the address of STA1.

NOTE

If you choose to have the TRA utility built with your system, each line is given one TRA unit used only for tracing activity on the line, loopback testing, or for reading error counters. The TRA unit is not polled by the host system and so should not be included in the number of stations that you specify in PEGEN.

Now, PEGEN asks you the same questions for BU-1. The host has assigned D1 as the address for the control unit corresponding to BU-1. You would therefore specify STA21 as the matching address for BU-1. For the three station addresses, you would specify STA0, STA1, and STA2 to correspond to the host addresses of hex 40, C1, and C2, respectively.

After completing the remainder of the PEGEN dialog, your PE system will be built. Each station on a controller is logically linked to a TU unit (pseudo device). Addresses given to the TU units correspond to the PE stations (or devices). The TU addresses are octal values, from 0 to the total number of stations recognized by the PE, minus one.

In the example above, the five stations would be assigned TU 0 through TU4. If you have the TRA option, one TRA unit is assigned to each line. The PE recognizes each TRA unit as a pseudodevice. Thus the total number of devices recognized by the PE increases according to the number of lines. You would then have seven TU units, two of which would be the TRA units. Each TRA unit is assigned the lowest TU address for its line. In the example above, the TRA unit for BU-0 would be assigned TU0, and the TRA unit for BU-1 would be assigned TU3. (Remember that the TU address assignments do not affect the host system's polling addresses.) Table 3-4 illustrates the assignment of TU addresses with and without the TRA program.

Appendix A provides examples of listings produced by PEGENs and further illustrates how controllers and stations are addressed and assigned TU units.

Table 3-4: How TU Devices Are Assigned to PE Stations and TRA Units

	WITHOUT TRA		WITH TRA	
	Device Symbol	**TU Device**	**Device Symbol**	**TU Device**
BU-0	STA0	TU0	TRA unit	TU0
	STA1	TU1	STA0	TU1
			STA1	TU2
BU-1	STA0	TU2	TRA unit	TU3
	STA1	TU3	STA0	TU4
	STA2	TU4	STA1	TU5
			STA2	TU6

Once controller and station addresses have been assigned, you may modify them by either doing a complete regeneration of the PE or by modifying the addresses contained in the PE.MAC configuration file (built during PEGEN), as explained in Section 6.6.4.

Chapter 4
The PEGEN Generation Dialog

To use PEGEN, you take part in a dialog with the PEGEN command file. Your responses are based on information that you gained during the planning phase.

This chapter describes the PEGEN dialog questions and tells you how to respond to them. Topics covered are:

- **PEGEN dialog display format.** How to respond to questions and how to get additional information about the questions.
- **PEGEN structure.** Description of the major sections of the dialog and of the numbering scheme used for PEGEN questions.
- **Determining the scope of the PEGEN dialog.** How to opt for either a complete or partial PE generation.
- **Starting PEGEN.** How to start the generation procedure.
- **The PEGEN dialog.** A listing of all the PEGEN questions and displays including explanatory material.

4.1 The PEGEN Dialog Display Format: Questions and Responses

The PEGEN dialog consists of:

- Questions (see section 4.1.1.)
- Optional explanatory text (see section 4.1.2.)
- Formats for responses to the questions (see section 4.1.3.)
- Default values applied in the event of non-responses (see section 4.1.4.)

4.1.1 Questions

In the PEGEN display each question is flagged with an asterisk (*) in the lefthand column to distinguish it from other PEGEN display information. Also, each question is numbered.

4.1.2 Optional Explanatory Text

When PEGEN asks a question, you have two options:

1. You can supply a valid answer.

2. You can press the ESCAPE key to display explanatory text which should help you answer the question. Explanatory text is not available for obvious or self-explanatory PEGEN questions.

For example, if you press ESCAPE in response to the question:

```
>* 1. Distribution device (DDNN)? [S] : <ESC>
```

PEGEN prints the following explanatory text and repeats the question:

```
>; Specify the device on which the distribution kit is
>; located (for example DK2). Optionally, the colon may
>; be supplied (for example, DK2:) The default device is
>; 'SY:'.

>* 1. Distribution device (DDNN) ? [S] :
```

4.1.3 Formats for Responses to the Questions

Your response to PEGEN questions will be one of three types:

- YES/NO
- Numeric
- Character string

The PEGEN dialog specifies the type of response that you must give, in square brackets ([ ]) following the text of the question.

Table 4-1 specifies the response types printed by PEGEN and the acceptable responses.

Table 4-1: Summary of PEGEN Responses

Response Form	PEGEN prompt:	You enter:
YES/NO	[Y/N]:	*Either:* 1. Y to indicate YES, or 2. N (or press (RET)) to indicate NO.
Numeric	[O R:*m*-*n*]:	An octal number in the range of *m* through *n*.
	[O R:*m*-*n* D:*d*]	*Either:* 1. An octal number in the range of *m* through *n*, or 2. Press RET to indicate that the default (*d*) should be used.
	[D R:*m*-*n*]	A decimal number in the range of *m* through *n*.
	[D R:*m*-*n* D:*d*]:	*Either:* 1. A decimal number in the range of *m* through *n*, or 2. Press RET to indicate that the default (*d*) should be used.
Character string	[S]:	A character string of any length.
	[S R:*m*-*n*]	A character string from *m* through *n* characters in length.

4.1.4 Default Values Applied in Event of Non-Responses

You can select the default response by pressing carriage return (RET). (PEGEN's default answer for a YES/NO question is NO.)

4.2 PEGEN Structure

PEGEN has five major sections of dialog:

1. **Introduction.** Notes and cautions are displayed.

2. **Preliminary.** You define characteristics of your target operating system and locate files and devices used during PEGEN.

3. **Component Selection.** You select the software components for your target system. Later, in the Component Parameters Section, you will specify the options for the components selected in this section.

4. **Configuration Database.** You specify system parameters, device characteristics, and communications buffers. This section generates the Configuration File (CETAB.MAC), which is the configuration data base used by the Communications Executive. Using the Configuration File Editor (CFE), you may change many of the configuration parameters after PEGEN.

5. **Component Parameters.** You specify partition information for all the components, and options for the utilities.

At the end of each section, signified by an (EOS), you are given the option of continuing on to the next section, pausing, redoing the section you just completed, or exiting from PEGEN. After a pause, you can continue PEGEN processing at the point at which you left off. After an exit, however, nothing is saved and you must restart at the beginning of PEGEN.

Except for the Component Selection section, these parts are divided into subsections. This logical segmenting of the PEGEN process is illustrated in Figure 4-1.

Question Numbering Scheme:

Each set of PEGEN questions associated with the lowest level of logical subdivision has a separate numbering sequence. For example, there are two numbering sequences used in the PEGEN Preliminary section: one for Target System Characteristics and one for PEGEN Procedure Parameters.

The numbers associated with the PEGEN questions in this chapter are also printed by PEGEN. This correspondence enables you to find the expanded description in this manual for any particular question issued during PEGEN.

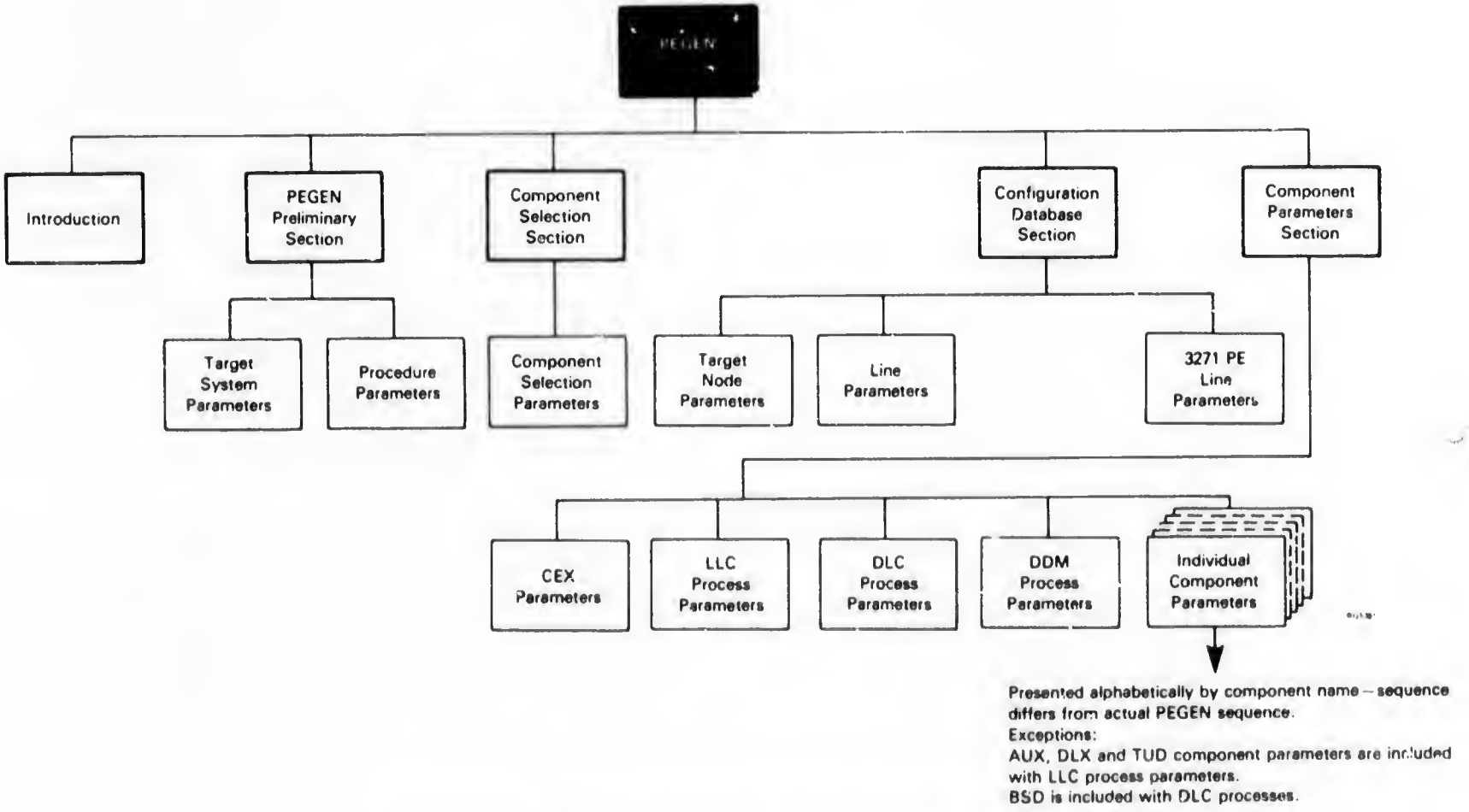


Figure 4-1: Divisions Within the PEGEN Process

4.3 Determining the Scope of the PEGEN Dialog

In the Component Selection section of the dialog you can choose a complete or partial generation of 3271 PE components and processes (question 1).

Choose the complete generation process if you are building a new system. PEGEN automatically selects the complete 3271 PE system.

Choose selected generation when modifying a system that you have already generated. PEGEN allows you to step selectively through the component selection process, asking you individual selection questions for each PE component, including the Communications Executive.

Example

If you want to add the Configuration File Editor to a system that you have already generated, answer the questions as follows:

Question no.	Question	Answer
>*1.	Complete 3271 PE system?	[Y/N]:N
>*10.	The Configuration File Editor (CFE)?	[Y/N]:Y

Answer N to all other selection questions.

Questions that will be asked in the Components Parameters section depend upon the responses you give in the Component Selection section.

(Table 4-2 lists the components of a PE system. The components are described in Section 2.2.)

Table 4-2: PEGEN Component Selection

Processes	Tasks	Other
	TRA (Trace program)	

The shaded components are the complete 3271 PE system generated with a YES response to question 1 of PEGEN's Component Selection section. TRA may be selected individually as an option with a YES response to question 2.

4.4 Starting PEGEN: Distribution Kits and Considerations

The following section describes the procedures for starting PEGEN and the preliminary considerations regarding the distribution kits containing your PE command files.

For Magnetic Tape Distribution Kits:

Use the command procedure PECOPY.CMD to copy the distribution kit to a disk.

Copy PECOPY.CMD from the distribution tape (*mmu:*) to a disk (*tdu:*) such as your system disk, and execute it using the following commands:

```
)UFD tdu:[137,10]                    ! Create UFD for PECOPY.CMD
)SET /UIC=[137,10]                   ! Set default UIC
)ALL mmu:                            ! Allocate tape drive
)MOU mmu:/FOR                        ! Mount distribution kit (M-PLUS only)
)FLX tdu:/RS/UI=mmu:[137,10]|PECOPY.CMD/DO/RW/DNS:bpi
                                     ! Copy PECOPY.CMD

)@tdu:PECOPY
```

Note that the disk *tdu*: must remain mounted throughout the PECOPY procedure, and that the PECOPY produced assumes that the distribution kit device is allocated and, if necessary, mounted.

Substitute the density of the distribution kit tape for *bpi*.

On an RSX-11M host system, do not issue the command 'MOU *mmu:*/FOR' because it is not supported by this system.

For Disk Distribution Kits:

To perform a PEGEN for an RSX-11M/M-PLUS system, enter the commands:

```
)MOU tdu:327108J

)@tdu:[137,10]|PEGEN
```

where

tdu: is the device on which the PEGEN distribution kit is loaded.

The PECOPY procedure can also be used to copy disk distribution kits to another disk. If you want to use the distribution kit disk to contain the output from PEGEN, you are advised to back up the distribution kit using PECOPY.CMD.

NOTE

RK05 distribution kits must be backed up before running PEGEN. The PEGEN procedure deletes unnecessary files from RK05 kits.

Make sure the distribution kit disk (*ddu:*) is write protected throughout the copying operation. To copy from the distribution kit disk (*ddu:*) to a disk, issue the following commands:

```
) ALL ddu:
) MOU ddu:3271OBJ
) SET /UIC=[137,10]
) @ddu:PECOPY
```

PECOPY dialog:

```
>; ================================================================
>;      * Copy the RSX-11/3271 PE Distribution kit *
>;                              Version 3.0
>; ================================================================
>;
>;
>; NOTE:
>;      This procedure assumes that the distribution kit is already
>;      allocated and, if necessary, mounted.
>;
>;      Explanations of any question can be obtained by answering
>;      the question with <ESC>.
>;
>* 1.   Do you want to copy a distribution kit?
>;
```

If you answer NO, PECOPY stops.

```
>* 2.   Distribution device (DDNN)?
>;
>; Specify the device on which the distribution kit is located (for
>; example DM2). Optionally, the colon may be supplied (for example,
>; DM2:). The default device is 'SY:'.
```

If you specify a magnetic tape device, PECOPY asks question 3 next; otherwise PECOPY asks question 4.

```
>* 3.   Is the distribution kit (PEkit) a 1600 bpi tape?
>;
```

```
>* 4.   Target disk (DDNN)?
>;
>; If you want to transfer the distribution files from magtape to
>; DB1: then answer 'DB1'. The default device is 'SY:'. This disk
>; will be your new distribution device!
>;
>; Note that the distribution kit can only be copied onto a disk.
```

If your target disk is already mounted, PEGEN asks no more questions.

```
>* 5.   Is tdu: ready?
>;
>; Please load your disk into drive tdu: and start the drive before
>; continuing.
```

```
>* 6.   Initialize the target disk (tdu:)?
>;
>; Answer Yes if you would like to have the disk initialized.
```

```
>* 7.   Volume label for tdu: [D:3271OBJ] ?
>;
>; A volume label is required on all mountable devices. Enter the
>; volume label for tdu:.
```

PECOPY now mounts the target disk unless it was already mounted.

```
>; TIME     PECOPY -
>;               Creating required UFDs on tdu:
>;
>;
>; TIME     PECOPY -
>;               Transferring the files from PEkit to tdu:
>;
>;
>; TIME      PECOPY -
>;               tdu: is your new distribution device
>;
```

4.5 PEGEN Dialog

All PEGEN dialogs — both descriptive text and questions — that you may encounter during PEGEN processing are given in this section. Also included are response instructions and supplementary explanations that are not part of the PEGEN displays. The conventions used are:

- Variable information is printed in italic. Variable information is based on your response to PEGEN questions and includes device names, partition names, and line designations.

- In PEGEN, references to network components and processes mean the software components that support the PE.

RSX-11M-PLUS and RSX-11M PEGENs are almost identical. Variations in procedures to be observed are indicated in the text.

Note that at the end of each section, PEGEN offers you the opportunity to continue, exit, pause, or repeat the section.

To perform a PEGEN for an RSX-11M/M-PLUS system, enter the following commands:

```
)MOU dev:32710BJ ! mount disk if necessary

)@dev:[137,10]PEGEN
```

where

dev: is the PEGEN distribution device.

After PEGEN is invoked, the introduction begins as follows.

The PEGEN Dialog

Introduction

```
NOTES AND CAUTIONS

>* 1.    Do you want to see the PEGEN notes/cautions? [Y/N]:
```

If you answer Y the following will be displayed:

```
>; Note the following:
>; ------------------
>;
>; * If you have not already performed a SYSGEN for your target
>;   system you must do so before continuing.
>;
>; * The privileged tasks generated during PEGEN use
>;   configuration dependant parameters produced by SYSGEN. For
>;   RSX-11M, the RSXMC.MAC file and the RSX11M.STB file
>;   for the target system are necessary. If they are not already
>;   on the output device under the proper UIC, PEGEN will move
>;   them from the user's system device to the output device.
>;
>; * If your distribution medium is the output device, it is
>;   recommended that you backup this device before continuing.
>;   This is recommended because the device must be write-enabled
>;   during the PEGEN sequence. If your disk is an RK05, certain
>;   files will be deleted to make sufficient space on the disk.
>;
>; * If you cannot always arrange to have the kit media and the
>;   target disk on the same drives, you should use the logical
>;   device assignments to ease PE maintenance. Four devices are
>;   required for the build -- a kit device, a target device, a
>;   listing device, and a map device.
>;
>;   For example, the following MCR commands create a typical set
>;   of logical names for the generation:
>;   . >ASN DK:=KI:
>;   . >ASN DB:=TG:
>;   . >ASN DB:=LS:
>;   . >ASN DB:=MP:
>;
>;   This assigns KI: as the kit device; TG: as the target
>;   device; LS: as the list device; and MP: as the map
>;   device. When asked for device names during PEGEN, use the
>;   assigned name for the respective device.
>;
>; * The proper system libraries for the target system must be on
>;   the LB: device under the UIC [6,1] while you are performing
>;   your PEGEN. These libraries include:
>;
>;      EXELIB.OLB
>;      EXEMC.MLB
>;      RSXMAC.SML
>;      SYSLIB.OLB
>;      NOANSLIB.OLB
>;
>;   Note that NOANSLIB.OLB is only required if your target
>;   system is RSX-11M-PLUS.
>;
```

```
>; * The slow task Builder (STK) is required by PEGEN. If this is
>;   not present on your system, see the RSX-11M/M-PLUS Task
>;   Builder Reference Manual for information on installing STK.
>;
>; * If tasks required by PEGEN are not installed on your host
>;   system, and you are logged in as a privileged user, PEGEN
>;   will install them automatically. The following tasks are
>;   required on your system:
>;
>;      PIP, LBR, MAC, TKB, and STK
>;      LOA and UNL          (if device drivers require loading)
>;      MOU and DMO          (if disks/tapes require mouting)
>;      UFD                  (if required UICs do not already exist)
>;      FLX                  (if target system files on magnetic tape)
>;
>; * For questions that can be answered YES or NO, NO is the
>;   default. For questions with defaults, the default is
>;   produced by pressing a carriage return (RET) in response to
>;   the question. Additional explanatory text for each question
>;   is available by hitting <ESC>.
```

At this point in order to satisfy the above requirements you may 'pause' or 'exit' the PEGEN procedure. If all requirements have been met you may continue with the procedure.

If PEGEN is unable to install the required system tasks, an error message is displayed and you are asked to specify the disk where the task images can be found:

```
>* 1. Are the task images on a mounted volume? [Y/N]: Y <ESC>
>;
>; The system tasks (PIP, MAC, TKB, LBR, UFD and STK) are required
>; by PEGEN. One, or more, of these tasks is not installed on your
>; host system, and PEGEN failed to find the task image files.
>;
>; If these tasks are available in [6,54] on a mounted
>; Files-11 volume then answer Y to this question.
>;
>; If you answer N, PEGEN will exit. Restart PEGEN when you are
>; ready


>* 2. Device for system tasks (DDNN)? [S]: DR0:
>;
>; Specify the device on which the system task image can be
>; found, for example DB1.
>; The Files-11 volume which contains the system task image files
>: in [1,54] must be loaded and mounted on this device.
```

```
>* 1. Target system type (M,M-PLUS)? [S]:
>;
>; The target system is the system for which you are
>; doing this PEGEN. The target system type determines
>; how the network software is to be generated and what
>; features are available.


>* 2. Distribution device (DDNN)? [S]:
>;
>; Specify the device on which the distribution kit is
>; loaded (for example,DK2). Optionally, the colon may be
>; supplied (for example, DK2:). The default device is SY:.
```

This establishes the location of the files needed for the rest of the generation.

```
                                    RSX-11M
>* 3. OK to delete unnecessary RSX-11M-PLUS files? [Y/N]:
>;
>; NOTE: An RK05 is insufficient to contain the entire kit
>;       throughout PEGEN. PEGEN will now delete all files
>;       unnecessary for the target system. If you do not have
>;       a copy of the kit, answer NO and make a copy before
>;       starting again.
```

If your target system is RSX-11M, the above question asks you if it is all right to delete unnecessary RSX-11M-PLUS files, and vice versa.

PEGEN gives the following information as it carries out the deletion:

```
>; Deleting unnecessary RSX-11M/M-PLUS files.
>;
>; Copying and renaming system-dependent files.
>;
>; The following are assumed:
>;
>;          Target system is (RSX11-M/M-PLUS).
>;          The system is mapped.
```

Target System Characteristics

```
>* 1. Processor type [S R:5-6]:
>;
>; Enter the target processor type, e.g, 11/34.


>* 2. (RESERVED)
>;


>* 3. Kernel data space enabled? [Y/N]:
>;
>; If your target system was generated with this feature,
>; answer YES to this question.
```

This question is asked in RSX-11M-PLUS generation dialogs only.

```
>* 4. More than 124K words of memory? [Y/N]:
>;
>; If your target system was generated with this feature,
>; answer YES to this question.
```

This question is asked only if the target processor type is a 22-bit machine (for example 11/70, 11/44)

```
>* 5. Multi-user protection? [Y/N]:
>;
>; If your target system was generated with this feature,
>; answer YES to this question.


>* 6. The extended instruction set (EIS)? [Y/N]:
>;
>; If your target processor includes this feature, answer YES to
>; this question.
```

This question is only asked for 11/35 and 11/40 target processor types.

```
>* 7. Powerfail recovery support? [Y/N]:
>;
>; If your target system was generated with this feature,
>; answer YES to this question.
```

This question is asked in RSX-11M generation dialogs only.

```
>* 8. Dynamic checkpoint allocation? [Y/N]:
>;
>; If your target system was generated with this feature,
>; answer YES to this question.
```

This question is asked in RSX-11M generation dialogs only.

PEGEN Procedure Parameters

```
>* 1. Target system device (DDNN)? [S]:
>;
>; This device will contain all generated and configuration dependant
>; files. *NOTE*-This must be a FILES-11 device.
>; The default device is SY:.


>* 2. Is target-system-device mounted? [Y/N]:
>;
>; If you have mounted target-system-device or if it has been
>; mounted as a public device, answer the question YES.
```

If you answer YES, question 6 is asked next.

```
>* 3. Is target-system-device ready? [Y/N]:
>;
>; Load your volume into drive target-system-device and
>; start the drive before continuing.
```

This question repeats until you specify that the device is ready.

```
>* 4. Volume label for target-system-device [S]:
>;
>; A volume label is required on all mountable devices. Enter
>; the volume label for target-system-device.
```

PEGEN attempts to mount the target system device; if it is unsuccessful question 1 is repeated.

```
>* 6. Group UIC for output? [0 R:1-377]:
>;
>; This is the group code under which all generated files will
>; reside. This feature will allow the user to do a number of PEGENs,
>; with outputs separated by UIC.
```

You may have several PEs generated under different UICs. However, only one PE can run at a time. Also note that once you select this UIC, you can change it only by another PEGEN. The UFDs are created on the output device by the generation procedure.

```
>* 7. Can the files be transferred by PIP? [Y/N]:
>;
>; The files RSX11M.STB and RSXMC.MAC are required. If the
>; input device is any FILES-11 device, then answer YES.
```

If you answer YES, PEGEN asks question 9 next.

```
>* 8. Can the files be transferred by FLX with a DOS format? [Y/N]:
>;
>; If the files are stored on tape in DOS format, answer YES.
>; Otherwise PEGEN will pause and ask you to transfer the files.
```

```
>* 9. What device are the system files on (DDNN)? [S]
>;
>; This is the storage device (for example, DX1) for the system
>; files: RSXMAC.MAC and RSX11M.STB. The default device is SY:.
```

The specified files belong to the target system on which the 3271 PE is to run. They do not necessarily belong to the system upon which the PE is being generated.

```
>* 10. Under which UIC is the file RSX11M.STB [X,Y]? (D:[1,54]) [S]:
>;
>; Specify where the file RSX11M.STB is located on LB: (for example,
>; [1,54]).
```

```
>* 11. Under which UIC is the file RSXMC.MAC [X,Y]? (D:[11,10]) [S]:
>; Where is RSXMC.MAC located on LB: (for example, [11,10]).
```

The RSX11M.STB and RSXMC.MAC files are located on the target system on which the PE is to run, not necessarily on the system where the PE is being generated.

```
>* 12. Listing device (DDNN)? [S]:
>;
>; The listing device should be either a FILES-11 device or LP:.
>; All listing files are generated on this device. By default,
>; no listing files are generated.
```

It is advisable to obtain listings, at least for the first generation. Direct the listings to a line printer if space is limited on your disks.

```
>* 13. Map device (DDNN)? [S]:
>;
>; The map device should be either a FILES-11 device or LP:. All map
>; files are generated on this device. By default, no map files are
>; generated.
```

```
>* 14. Spool listings and/or maps? [Y/N]:
>;
>; If you want the listing and/or the maps spooled, answer YES.
```

```
>* 15. Delete build files after use? [Y/N]:
>;
>; It is recommended that you have at least 1000 free blocks
>; available on the output device, if you choose not to delete.
>; An additional 600 blocks should be available if maps and
>; listings are desired.
```

If you choose to delete the build files after use, you will conserve disk storage space. However, it will be difficult to rebuild an individual component if a change is necessary.

COMPONENT SELECTION

```
>* 1. Do you want the complete 3271 system? [Y/N]:
>;
>; Answer YES to get a complete 3271 PE system. This includes
>; the Communication Executive; the processes: TUD, AUX, BSD, and
>; DDMs; and the utilities: NCP, NTL, NTINIT, VNP, CFE, NDA, DCE,
>; LBK.
>;
>; Note: Trace is required for LBK. If Trace is not selected, your LBK
>; program will not function properly.
```

On an RSX-11M-PLUS system, DLX (Direct Line Controller) is included and VNP is not included.

If you answer NO to 1 above, PEGEN asks questions 2 to 14. If you answer YES to 1, PEGEN asks only questions 2, 13, and 14. See the discussion on selecting PE components, Section 4.4.

```
>* 2. Do you want the Trace program? [Y/N]:
>;
>; The Trace program (TRA) provides a realtime trace of the
>; activity on a 3271 line and formats the activity for later study.
>; It is useful for debugging and for seeing the actual line activity
>; as seen by the protocol emulator. It formats all characters
>; collected by the traced unit both received from the master and
>; data and protocol messages generated by the user or the protocol
>; emulator. Trace must be selected for LBK to work properly.
```

The trace feature allows you to write a program, monitor activity on the line, and analyze the transmission from the IBM host before this transmission has been manipulated by the PE.

The Trace unit is added to each line as the lowest TU unit of the line. The Trace unit is not addressable by the IBM system. Section 3.5 explains how Trace units are configured and addressed, and Section 8.2.3 explains what Trace does and how you can use it.

```
>* 3. Do you want the DCE program? [Y/N]:
>;
>; DCE is the process that reads and resets the statistics generated
>; by the BSD process for each active line. It then formats them and
>; puts them into a file for later access by the system manager.
```

If you answered YES to 1, PEGEN does not ask this question.

The DCE Error Counter Report indicates whether the host is active (that is, whether the PE is being polled by the host) and provides such counts as the number of data blocks transmitted and received; the number of NAKs, transmitted and received; the number of 3270 command rejects, and the number of hardware errors. See Figure 6-1 for a sample DCE Error Counter Report.

```
>* 4. Do you want the LBK program? [Y/N]:
>;
>; LBK is a utility program that allows the system manager to test
>; 3271 PE lines. It offers loopback, modem sense, and line monitor
>; functions.
```

If you answered YES to 1, or NO to 2, PEGEN does not ask this question. If you answered NO to 1, then PEGEN asks certain logically related questions from 5 through 11. Otherwise, PEGEN asks question 13 next.

```
>* 5. The Communications Executive? [Y/N]:
>;
>; The Communications Executive (CEX) provides the basic functions of
>; interfacing and scheduling of other network processes, timer
>; services and resource management of common buffer pools.
```

If you answer YES to question 5, PEGEN builds the processes, loader, and SCR utilities as well.

```
>* 6. The processes? [Y/N]:
>;
>; The processes include the drivers, AUX, TUD, BSD, and DLX.
```

The processes include the drivers (BU, KBU, and DV), BSD, AUX, TUD, and DLX. If you answer YES to this question, you are also asked questions in the Configuration Data Base section. Note that you cannot select individual processes.

```
>* 7. The Network-Loader/Initializer (NTL, NTINIT)? [Y/N]:
>;
>; The Network-Loader/Initializer is needed to initialize the network
>; by loading the Communications Executive and all processes that
>; are marked for load. If a line is set for ON, the line state is
>; set to ON by the loader.

>* 8. The Network Control Program (NCP)? [Y/N]:
>;
>; The Network Control Program (NCP) is a utility program that
>; accepts terminal commands to load, control, monitor and test
>; the 3271 PE software.

>* 9. The Virtual Network Program (VNP)? [Y/N]:
>;
>; The Virtual Network Program (VNP) performs control functions,
>; similar to those of NCP, on the disk image of an RSX 3271 PE
>; system.
```

The above question does not appear with RSX-11M-PLUS systems. (VNP is not available on RSX-11M-PLUS systems.)

```
>* 10. The Configuration File Editor (CFE)? [Y/N]:
>;
>; The Configuration File Editor (CFE) allows the user to change
>; dynamic information in the configuration data base, such as the
>; vector and CSR addresses.


>* 11. The Network Dump Analyzer (NDA)? [Y/N]:
>;
>; The Network Dump Analyzer performs crash dump analysis for the
>; network data structures in the same way that CDA analyzes the
>; operating system data structures.


>* 12. The Screen Emulator (SCR)? [Y/N]:
>;
>; The Screen Emulator can be used to enter data as though on
>; a 3270 series terminal.


>* 13. Should all tasks be built for the same partition? [Y/N]:
>;
>; If all of the network tasks are to be built to run in the
>; same partition, you may enter the name of the partition here and
>; reduce the number of questions that will be asked during PEGEN.
```

Network refers to the software components that support the PE.

```
>* 14. What is the partition name? [S R:0-6]:
>;
>; Enter the name of the partition to be used by the network tasks.
>; The default name (RET) is GEN.
```

Network refers to the software components that support the PE. If you selected the complete 3271 PE system, or the Communications Executive, or the processes, the Configuration Database section of PEGEN appears next. Otherwise, the Component Parameters section appears next.

Database Parameters

```
>* 1. Number of control buffers (CCBs)? [D R:4.-200]:
>;
>; Control buffers (Communications Control Blocks and CCBs) are used
>; by network processes to communicate with and schedule each other.
```

The 3271 PE uses CCBs for interprocess communication. Each line requires three CCBs when in the on state and two extra to initialize; an additional CCB is needed for each user to read or write. If you wish to queue several receives and transmits to each unit, specify a large number of CCBs, but be careful: CCBs require system dynamic memory. You can adjust the number of CCBs later (by using CFE) to tune system performance. See the discussion of buffer allocation considerations in Section 3.4.

```
>* 2. Number of large buffers (LDBs)? [D R:4.-200.]:
>;
>; Large buffers (LDBs) are used for the transmission and
>; reception of data.
```

See discussion of buffer allocation considerations in Section 3.4. You can change the number of LDBs at a later time by using CFE's DEFINE SYSTEM command. See Section 6.6.3.2.

```
>* 3. Large buffer (LDB) size? [D R:192.-1024. D:280.]:
>;
>; The 3271 PE can use multiple LDBs to buffer user data.
>; However, on a UMR mapped system with a DV or KBU, the minimum
>; LDB size is 254 bytes. The LDB size will affect performance.
>; Choosing the LDB size involves a trade-off between available
>; memory and system performance. Since the 3271 PE needs a
>; certain number of buffers per line, you should choose a
>; large enough LDB to hold typical messages, yet leave enough
>; pool to have approximately two LDB's per line.
```

See discussion of buffer allocation considerations in Section 3.4. You can change the LDB buffer size later by using CFE's DEFINE SYSTEM command. See Section 6.6.3.2.

```
>* 4. Number of large buffers reserved for receives?
>; [D R:1.-29.D:3.]:
>;
>; Large buffers (LDBs) are used for both transmitting and
>; receiving data segments. If the pool size falls below the
>; threshold value, then only receive buffer allocation
>; requests will be honored.
```

If you have chosen to build tasks to run in the same partition (you answered YES to question 13 of the Component Selection section), then question 8 appears next. Otherwise PEGEN asks question 5 next.

```
>* 5. Create the network buffer pool partition dynamically? [Y/N]:
>;
>; Answer YES to this question if you would like the network
>; buffer pool to be dynamically allocated when the network is
>; loaded; answer NO if you already have a dedicated common block
>; partition in the target system. If the pool is to be dynamically
>; allocated, it will be allocated from a system controlled
>; partition.
>;
>; NOTE: Answering YES to this question enables the user to
>;       change the number and/or size of the buffers without
>;       having to repartition the system.
```

If you answer NO to question 5, PEGEN asks question 6 next and then starts the next section. If you answer YES to question 5, PEGEN asks question 7 next and then question 8.

```
>* 6. Network buffer pool partition name? [S R:0-6]:
>;
>; Enter the name of the partition in which the network buffer
>; pool is to reside. This must be a common partition.


>* 7. System controlled partition name for pool? [S R:0-6]
>;
>; Enter the name of the system controlled partition from
>; which the network buffer pool is to be allocated.
>; The default partition name is GEN.


>* 8. Should the network be top-down loaded in partition-name?
>; [Y/N]:
>;
>; Answer YES to this question if you want the network software
>; to be allocated out of the top of partition-name.
>; This helps prevent partition fragmentation.
```

You can change this specification later with CFE.

Line Parameters

```
>* 1. Device Driver Process name [S R:0-3]:
>;
>; The Device Driver Process (DDM) names correspond to
>; the two or three character device name. Enter a
>; carriage return when there are no more names.
>;
>; Legal DDM process names are:
>;
>; Process Name          Type     MUX?        DMA?
>;
>; BU                    Sync     No          No
>; KBU                   Sync     Yes         Yes
>; DV                    Sync     Yes         Yes
```

To use the KBU, you must have a KMC-11A and at least one DUP-11 on the system since there is no direct connection between the KMC-11A and the DUP-11 other than the UNIBUS. The DUP-11 must be REV. J or later. It is recommended that you generate both a KBU driver and the standard BU driver for your DUPs. This way, if the KMC should fail for any reason, you can switch back to running the DUPs using the BU driver.

The device driver that you specify determines which questions will follow. After you answer all questions for each device driver, PEGEN returns to question 1. When you have specified all device drivers, enter a carriage return in response to question 1 and PEGEN advances to the next subsection.

```
>* 2. Number of process-name controllers? [D R:1.-16. D:1.]:
>;
>; Enter the number of process-name controllers that are
>; installed on the target system. Do not include any controllers
>; which were specified during target system SYSGEN, since devices
>; cannot be shared between the PE and the operating system.


>* 3. Are all process-names installed at the same device
>; priority? [Y/N]:
>;
>; All peripheral interfaces on PDP-11 processors are installed
>; with a device priority in the range of 4 - 6 which, together
>; with its position on the UNIBUS, determines its priority in
>: arbitrating control from the UNIBUS. Answer YES to this
>; question if all process-names have been installed at the
>; same hardware device priority level.


>* 4. What is the process-name device priority
>; [0 R:4-6 D:5]:
>;
>; Enter the hardware device priority level of all process-names
>; on the target system.
```

```
>* 5. Number of lines? [D R:1.-48.D:1]:
>;
>; For multiplexer devices, respond to this question with
>; the number of lines for controller process-name-n.
```

Answer this question and the next two with the hardware setting of the (process-name) device driver.

```
>* 6. Vector address? [O R:0-774 D:0]:
>;
>; If the interrupt vector for process-name-n is not known,
>; answer this question with a 0. The vector address may be
>; overridden using the NCP or VNP utility when the line is loaded.
```

The interrupt vector for the process-name is a floating vector. If you do not know the vector address, contact your local Field Service representative. If you do not specify the vector address during system generation, you can do so with the NCP SET LINE command at installation.

```
>* 7. CSR address? [O R:160000-177777 D:177777]:
>;
>; If the device control and status register (CSR) address for
>; process-name-n is not known, answer this question with a
>; 177777. The CSR address can be overridden using the NCP or
>; VNP utility when the line is loaded.
```

The CSR address for the process-name is part of the floating address space. You can also specify the CSR address with the NCP or VNP utility, using the SET LINE command. The CSR address can also be specified with the CFE utility using the DEFINE LINE command.

If you answered NO to question 3, PEGEN asks question 8 next; otherwise it asks question 9.

```
>* 8. Device priority? [O R:4-6 D:5]
>;
>; Enter the hardware device priority for process-name-n.
>; The priority must be in the range 4 to 6.
```

PEGEN asks questions 9 to 11 for each of the lines. You specified the number of lines in question 5 of this PEGEN section.

```
>* 9. Unit CSR address? [O R:160000-177777 D:177777]:
>;
>; For the KBU device, you should enter the CSR address of the
>; DUP-11 which the KMC-11 is controlling.
```

PEGEN asks question 9 only for lines associated with the KBU device.

```
>* 10. Number of stations? [D R:1.-32.]:
>;
>; Respond to the question with the number of device station
>; addresses that process-name-n-ln will be polled for,
>; by the host system.
```

Do not count trace units; they are not recognized by the host system.

```
>* 11. Set line state to ON on loading the network? [Y/N]:
>;
>; If process-name is set to ON, the process-name device
>; driver (DDM) and the line's database will be automatically
>; loaded into memory at network initialization time. The line
>; state will then be set ON and the network software
>; (once initialized) will attempt node initialization on
>; that line. If it is not enabled, the user may set the state
>; of the line to ON with either the NCP or VNP utility.
>; If the line is multipoint, all tributaries on
>; the line will also be set to ON.
```

NOTE

The IBM host must be polling your system when you turn the line ON. Check this before you use NCP to set the state of the line to ON. If you have a dial-up line, always answer NO to question 11.

```
>* 12. Controller address for process-name-n? [S R:4-5]:
>;
>; This question sets the controller address that this line
>; will recognize during polling by the host system. The
>; controller address should be specified with the symbol
>; STAnumber, where STAnumber is the symbol on the same row
>; with the correct address character in the following table:
>;
```

CU or Device Symbol	EBCDIC I/O Char.	Hex	Octal	CU Number Octal	Cu Number Decimal
STA0 SP	40	100	0	0	
STA1	A	C1	301	1	1
STA2	B	C2	302	2	2
STA3	C	C3	303	3	3
STA4	D	C4	304	4	4
STA5	E	C5	305	5	5
STA6	F	C6	306	6	6
STA7	G	C7	307	7	7
STA10	H	C8	310	10	8
STA11	I	C9	311	11	9
STA12	CENT	4A	112	12	10
STA13	.	4B	113	13	11
STA14	<	4C	114	14	12
STA15	(	4D	115	15	13
STA16	+	4E	116	16	14
STA17	BAR	4F	117	17	15
STA20	&	50	120	20	16
STA21	J	D1	321	21	17
STA22	K	D2	322	22	18
STA23	L	D3	323	23	19

```
>; STA24          M     D4     324     24     20
>; STA25          N     D5     325     25     21
>; STA26          O     D6     326     26     22
>; STA27          P     D7     327     27     23
>; STA30          Q     D8     330     30     24
>; STA31          R     D9     331     31     25
>; STA32          !     5A     132     32     26
>; STA33          $     5B     133     33     27
>; STA34          *     5C     134     34     28
>; STA35          )     5D     135     35     29
>; STA36          ;     5E     136     36     30
>; STA31          ¬     5F     137     37     31
>;
>; The answer for station 0 would be STA0.
```

Specify the controller address as STAnn where STAnn is one of the symbols indicated in column 1 of the table above. These symbols correspond to the physical CU or device number. See Section 3.5 for a discussion of Control Unit addressing and for considerations to be taken in responding to this question. Controller addresses may be modified later by performing a complete regeneration of the PE or following the procedures outlined in Section 6.6.4.

```
>* 13. Station address for process-name-n station x Unit
>; TUx:? [S R:4-5]:
>;
>; This question is asked for each station on the controller.
>; You will be asked for x unique STAnumbers for this controller.
>; Please answer the question as for the controller address
>; (for example, STA0).
```

Trace units are not considered here. The station address may be modified later by performing a complete regeneration of the PE or following the procedures outlined in Section 6.6.4.

```
>* 1. Number of NAKs received before abort? [D R:3.-30.D:7.]:
>;
>; Specify the number of NAKs that can be received by the
>; 3271 PE before the host system aborts transmission of the
>; current message segment (at which point you will be
>; informed of an error condition).
```

This is a protocol matter. The host system responds with a Negative Acknowledgement (NAK) when there is a communication failure on the line or a protocol error in the message. If the number of NAKs received significantly exceeds the number you specify here, you can assume that there is either a communication problem or a hardware failure. The 3271 PE will abort the current transaction, send an End of Transmission character (EOT), and increment the NAK thresholds-exceeded counter of the DCE error logging routine (see Section 2.2.6).

```
>* 2. Number of ENQs to be sent on error? [D R:3.-30.D:7]:
>;
>; Specify the number of ENQs to be sent by the 3271 PE to the
>; host system if a receive time out should occur or a receive
>; error occurs for a protocol message.
```

The 3271 PE transmits an ENQ (ENQuiry) upon failure to receive the correct protocol response from the master station. An incorrect response may be due to a master or communications failure. Excessive ENQs indicate a problem which should be diagnosed and solved. When the number of ENQs exceeds the specified number, the PE increments the DCE acknowledgement timeout thresholds-exceeded counter.

```
>* 3. Number of SYNs on messages? [D R:3.-15.D:3.]:
>;
>; This is the number of SYNs to be sent before each data and
>; protocol message. It may be necessary to increase the number
>; to compensate for bad lines or to delay message transmission.
>; The latter is desirable if, for example, the host system is
>; slow in posting a read for the 3271 PE's responses.
>; The protocol requires a minimum of three.
```

A sequence of SYNs (SYNchronous Idles) is needed to establish transmitter/receiver synchronization. The number of SYNs needed may vary depending on communications equipment. If you are in any doubt, check the number of SYNs transmitted by the IBM host and use that number. As a rule, it is safer to have too many SYNs than too few.

COMPONENT PARAMETERS

The Communications Executive (CEX) provides interfacing and scheduling support for other components, general timer service, and resource management of common buffer pools. The following questions define the Communications Executive. Note that any space allocated in the CEX partition (CEXPAR) in excess of the amount needed for resident code will be used for CCBs.

Communications Executive (CEX)

The following question appears only if your target system is RSX-11M-PLUS. Otherwise questions 2 to 4 appear.

```
>* 1. Value of $CEX in RSX-11M-PLUS EXEC? [O R:20000-120000]:
>;
>; The Communications Executive provides:
>;
>; . Interprocess communications and scheduling services
>; . General timer service for communications processes
>; . Resource management of common buffer pools
>;
>; Space for the Communications Executive was generated
>; within the RSX-11M-PLUS Executive when you selected
>; DECnet support during SYSGEN. Enter the value of the
>; label $CEX from the RSX-11M-PLUS Executive map.

>* 2. Communications Executive's partition name? [S R:0-6]:
>;
>; The Communications Executive provides:
>;
>; . Interprocess communications and scheduling services
>; . General timer service for communications processes
>; . Resource management of common buffer pools
>;
>; Enter the name of the partition where the Communications
>; Executive (CEX) is to be loaded. The default partition
>; name is CEXPAR. This partition must be in the RSX
>; Executive's address space.

>* 3. partition-name partition base address? [O R:14000-117000]:
>;
>; Enter the base address (in octal) of the partition-name
>; partition on the target system. The Communications
>; Executive must reside below 120000 on 20K-Executive
>; systems and below 100000 on 16K-Executive systems.

>* 4. partition-name partition length? [O R:2700-6000]:
>;
>; Enter the length (in octal) of the (partition-name)
>; partition on the target system.

>* 5. Does the target system have a KG-11 [Y/N]:
>;
>; The KG-11 communications arithmetic option computes
>; the cyclic redundancy check characters which are used
>; for error detection in serially transmitted/received data.
>; If the target system does not have the KG-11 option
>; and the communications interface option(s) installed on the
>; target system CRC checking, a software algorithm will be used.
```

This option is useful only on a system with DUP11s (since the DV11 and KBU hardware performs its own CRC checking).

If you are building tasks for the same partition (your response to question 13 of the Component Selection section was YES), then most of the following questions are not asked.

LLC Processes

AUX Parameters

```
>* 1. Partition name [S R:0-6]:
>;
>; Enter the name of the partition in which the AUX process
>; is to be loaded. The default name (RET) is GEN.
```

DLX Parameters

```
>* 1. Partition name [S R:0-6]:
>;
>; Enter the name of the partition in which the DLX process
>; is to be loaded. The default name (RET) is GEN.
```

TUD Parameters

```
>* 1. Partition name? [S R:0-6]:
>;
>; Enter the name of the partition in which the TUD process
>; is to be loaded. The default name (RET) is GEN.


>* 2. Maximum transmission block size? [D R:100.-1920. D:280.]:
>;
>; This question asks for the maximum transmit block size
>; for the protocol emulator. This will set the maximum
>; block size to the value given except on a UMR mapped
>; system with a DV or KBU. In those systems, the maximum
>; transmit block size is dynamically set to the receive
>; data buffer size in CETAB. For all other configurations
>; the value given here is the maximum before the application
>; gets an IE.RBG error on a write QIO.
```

DLC Processes

BSD Parameters

```
>* 1. Partition name? [S R:0-6]:
>;
>; Enter the name of the partition in which the BSD process
>; is to be loaded. The default name (RET) is GEN.


>* 2. Does this system support the TRACE utility? [Y/N]:
>;
>; Answer YES if the Trace feature was selected for this system.
```

DDM Processes

The questions that appear in this section depend upon your selections in the Configuration Database section.

BU Parameters

```
>* 1. Driver partition name? [S R:0-6]:
>;
>; Enter the name of the partition in which the BU driver is
>; to be loaded. The default name (RET) is GEN.
```

DV Parameters

```
>* 1. Driver partition name? [S R:0-6]:
>;
>; Enter the name of the partition in which the DV driver is
>; to be loaded. The default name (RET) is GEN.
```

KBU Parameters

```
>* 1. Driver partition name? [S R:0-6]:
>;
>; Enter the name of the partition in which the KBU driver is
>; to be loaded. The default name (RET) is GEN.
```

NTINIT (Network Initializer)

```
>* 1. Does your system have a BISYNC KMC/DUP (KBU)? [Y/N]
>;
>; Answer YES to this question if your configuration includes
>; a KMC/DUP (KBU). If your system does not have a KBU, the
>; size of NTINIT can be reduced because no microcode is required.


>* 2. Partition name? [S R:0-6]:
>;
>; Enter the name of the partition where the Network Initializer
>; is to be loaded. The default name (RET) is GEN.
```

Network Loader (NTL)

```
>* 1. Partition name? [S R:0-6]:
>;
>; Enter the name of the partition where the Network Loader is
>; to be loaded. The default name (RET) is GEN.
```

Network Control Program (NCP)

```
>* 1. Partition name? [S R:0-6]:
>;
>; Enter the name of the partition where the NCP task is
>; to be loaded. The default name (RET) is GEN.
```

Configuration File Editor (CFE)

```
>* 1. Partition name? [S R:0-6]:
>;
>; Enter the name of the partition where the Configuration File
>; Editor (CFE) is to be loaded. The default name (RET) is GEN.
```

Line Trace Utility (TRA)

```
>* 1. Partition name [S R:0-6]:
>;
>; Enter the name of the partition where the Trace Utility is
>; to be loaded. The default name (RET) is GEN.
```

Device Counters Utility (DCE)

```
>* 1. Do you wish the log file to be 'SY:[target UIC]TULOG.TXT'?
>; [Y/N]:
>;
>; Do you want the DCE logfile to be 'SY:[target UIC]TULOG.TXT'?
>; If you wish it to be something else, answer NO to
>; this question. Then you will be asked for:
>;
>; 1. A device to replace SY:
>; 2. A UIC in the form X,Y (e.g. 301,54), and
>; 3. A file name in the form NAME.EXT (e.g. TULOG.TXT)
```

If you answer NO to question 1, questions 2 to 4 appear; otherwise, question 5 may appear next.

```
>* 2. Log device in the form DDNN: [S]:
>;
>; Enter the device on which you want the DCE log file
>; to be created, in the form DDNN:. The default
>; log device is SY:.


>* 3. UIC in the form X,Y (e.g. 210,54) [S]:
>;
>; Enter the UIC where you want the DCE log file to be
>; created, in the form x,y. The default UIC is target UIC.


>* 4. File name in the form NAME.EXT (e.g. TULOG.TXT) [S]:
>;
>; Enter the name you want to be used for the DCE log file,
>; in the form NAME.EXT. The default log file name is TULOG.TXT.
```

Question 5 appears only if you answered NO to question 13 of the Component Selection section.

```
>* 5. Partition name? [S R:0-6]:
>;
>; Enter the name of the partition where the DCE utility is
>; to be loaded. The default name (RET) is GEN.
```

Loopback Utility (LBK)

```
>* 1. Partition name? [S R:0-6]:
>;
>; Enter the name of the partition in which the LBK utility
>; is to be loaded. The default name (RET) is GEN.
```

Network Dump Analyzer (NDA)

```
>* 1. Partition name? [S R:0-6]:
>;
>; Enter the name of the partition where the NDA utility
>; is to be loaded. The default name (RET) is GEN.
```

Screen Emulator

```
>* 1. Partition name? [S R:0-6]:
>;
>; Enter the name of the partition in which the SCR utility
>; is to be loaded. The default name (RET) is GEN.


>* 2. Number of read buffers ? [2 R:3.-15. D:3]:
>;
>; Enter the number of read buffers to be used by the SCR utility.
>; The default value is 3. The number of buffers is dependent on
>; the application running but the default value is usually
>; applicable unless you expect the host system to send large
>; blocks of unsolicited data. For example, a value of 14 is
>; more suitable if you are running under VM. Note that the size
>; of SCR is increased by about 2K bytes for each additional
>; read buffer.


>* 3. First TU unit available for SCR [0 R:1-h D:1]:
>;
>* 4. Highest TU unit available for SCR [0 R:1-h D:h]:
>;
>; Specify the range of TU units that can be used by the
>; SCR utility
>;
>; By default, the utility will select any free TU unit in
>; the full range available, (TU1 through TUh)
>;
>; If you wish the selection to be limited, specify the first
>; and last units in a range. For example, if you wish to reserve
>; some TU units for specific applications, specify a range
>; that does not include these units.
```

TU*l*=lowest TU unit available
TU*h*=highest TU unit available

Questions 3 and 4 are always asked together

Virtual Network Program (VNP)

```
>* 1. Partition name? [S R:0-6]:
>;
>; Enter the name of the partition where VNP
>; is to be loaded. The default name (RET) is GEN.
```

Building 3271 PE components

```
>; All questions have been answered. PEGEN will now proceed to
>; build your system the way you have chosen. This can take
>; anywhere from 5 minutes to 3 hours, depending on your
>; selections and your host system configuration.
```

If your system crashes, or you need to stop PEGEN for other reasons, you can restart the build procedure as follows:

)@*dev:*[*x*,24]BLDNET

where

dev: is the target disk

x is the User Identification Code (UIC) group number that was specified for the 3271 Protocol Emulator (PE) to be built in.

Make sure that all media are mounted in the same devices as during the original PEGEN, before you restart the command file to build the 3271 PE. It is recommended that you do not specify that build files be deleted.

To build individual components for the system, use the following command:

)TKB @*dev:*[*x*,24]*com*BLD

where

dev: *and x* are as above

com is the component name (e.g. DV, VNP)

All questions answered, PEGEN will now begin to build your system. The process is automatic and can run unattended. PEGEN lists the build proceedings as they occur. When the procedure completes, review the output to verify the completeness of the PE you produced with PEGEN.

See Chapter 5 for details on installation and verification procedures. See Appendix A for sample PEGEN listings.

Chapter 5
Installation and Verification

Chapter 4 described how to configure and build a PE. The following procedures, described in this chapter, enable you to load and test the software:

1. Procedures for loading the software (Section 5.1).

2. Several post-installation procedures that help you determine that the PE functions properly in an operational setting (Section 5.2).

All procedures specified in this chapter apply equally to a PE under RSX-11M or M-PLUS.

5.1 Installing, Loading, and Starting

After PEGEN has successfully completed, you can install, load, and start up the PE by means of the three-step procedure described in this section.

This procedure can be used to install, load, and start up a 3271 PE.

NOTE

Only those lines that were marked for load in PEGEN will be ON when the system is brought up. To turn on the lines that were not marked for load in PEGEN, use the appropriate NCP, VNP, or CFE command. See Chapter 6.

Step 1. Invoke and execute an indirect command file called NETINS by entering:

@[*netuic*,24]NETINS

where *netuic* is the user identity code specified in question 14 of the PEGEN Procedure Parameters section of PEGEN. The following question then appears:

)* On what device are the network tasks [S]:

Step 2. Respond to the above question with the name and number of the device on which the PE resides. The following question then appears:

)* Do you want to load the network? [Y/N]:

If you do not want to load the system you have built, respond N. The tasks will then be installed, but they will not be loaded.

If you want to load the system you have built, respond Y. The system will be loaded. Each step of the load procedure is logged on your terminal as illustrated by the sample listing in Figure 5-1.

```
>@NETINS/NOMCR
> * On what device are the 3271 PE tasks [S]:
> * Do you want to load the 3271 PE? [Y/N]:
>!ASN SY:=XX:
>INS XX:[370,54]NTINIT
>INS XX:[370,54]NTL
>INS XX:[372,54]NCP
>INS XX:[370,54]VNP
>INS XX:[370,54]CFE
>INS XX:[370,54]NDA
>INS XX:[370,54]SCR
>INS XX:[370,54]TRA
>INS XX:[370,54]DCE
>INS XX:[370.54]LBK
>NCP SET SYS
>ASN =XX:
>@ <EOF>
```

Figure 5-1: Install, Load, and Start Procedure for PE: Sample Listing

Step 3. Connect to the IBM host.

5.2 Post-Installation Verification

This procedure enables you to check an installed 3271 PE and verify that it is running and receiving poll messages from the host system. This procedure checks the PE step-by-step to verify that each level of the system is functioning properly. As part of the verification procedure, you will be running the Loopback Test Program (LBK). Note that this program runs properly only if the Trace option has been selected after PEGEN.

Step 1. Install and load the PE as described in Section 5.1.

Step 2. Turn on the PE lines to be tested by entering a command similar to the following:

)NCP SET LIN BU-*n* STA ON

where

n is the line number

Step 3. Manually set the line interface into a loopback configuration. This is done by attaching a DEC loopback test connector (H325) to the cable or by placing your modem in a loopback state (see Figures 8-4 and 8-5). Be careful not to exceed the EIA specifications while looping back. Note that the passive loopback connector, like the H325, requires that the signals from the line interface travel twice the distance of the cable. Section 8.2.4 details the Loopback Test procedure

Step 4. Run the Loopback Test Program (LBK) to test:

1. The software for the line being checked. (See Section 5.2.1.)

2. That the test message looped back is identical to the test message transmitted. (See Section 5.2.2.)

3. That the PE is receiving polls from the host. (See Section 5.2.3.)

NOTE

Before proceeding with the test, read Section 8.2.4 to understand what the Loopback Test Program does and how it is to be used.

5.2.1 Testing Line Software

Use LBK's SHOW command to display the status of the modem control signals. Problems at this point indicate that the line is not in the right state or the software that supports the line is not configured properly. Check the current state of the communications line, the configuration file, using CFE, and your responses to the PEGEN dialog. Check the latter two items to see if your PE was generated correctly for your hardware configuration. For example, did you enter the correct device type and CSR vector? Did you select Trace? Correct any problems and try again.

Example. Using LBK's SHOW command:

Show the status of the modem control signals on the first line interface by selecting the Trace unit, (TU0:). Enter:

```
LBK)SHOW/LINE=0
```

An example of a display that may appear in response to this command is:

```
Modem status
  DTR=ON  DSR=ON  RTS=OFF CTS=OFF
  DTR=ON  DSR=ON  RTS=OFF CTS=OFF
  DTR=ON  DSR=ON  RTS=OFF CTS=OFF
Test complete
```

5.2.2 Testing Line Transmission/Reception Capability

Use the LOOP command to check the capability of the line. Select the Trace unit as the line to test on each line interface. The LOOP command sends a data pattern on the specified line and expects to receive the same data back.

If all goes well you should always get the 'Normal completion on transfer' message on each transfer. If there are problems, a message will appear indicating that errors are occurring. Check the line connections for proper electrical contact and try again. If you have transmit timeout problems, check to see if you have a properly set synchronous clock.

Example. Using LBK's LOOP command:

Check the transmission and reception capability of the first line interface by selecting the Trace unit, (TU0:). Enter:

```
LBK) LOOP/LINE=0
```

An example of a display that may appear in response to this command is:

```
Loopback test starting
Normal completion on transfer
Normal completion on transfer
Normal completion on transfer
Test complete
```

When you have finished checking the line interface, disconnect the loopback and connect the cable to the host system.

5.2.3 Verifying Proper Reception of Host Polls

After testing the line by checking the software support and modem control status, use the Loopback Test Program MONITOR command to verify that the system is receiving polls from the host system. The MONITOR command displays in hexadecimal any data received on the line from the host.

Example. Using LBK's MONITOR command:

To check if your first line interface is receiving polls from the host system, select the Trace unit (TU0:). The ITERATIONS switch is used to specify the number of times that the line is to be tested:

```
LBK) MONITOR/LINE=0/ITERATIONS=4
```

The example dislayed below shows that this line is receiving polls to controller address STA0, which corresponds to hexadecimal 40, the address known to the host (see Section 7.3.1 for a description of the polling message format):

```
 Monitor mode
R 40407F7F2D
R 40407F7F2D
R 40407F7F2D
R 40407F7F2D
 Test complete
```

The 3271 PE has now completed its verification.

Chapter 6
Operating and Maintaining the Configuration

This chapter describes the utility commands that you can use to perform the following functions:

1. Load and unload PE software (using NCP).
2. Start up and shut down the PE (using NCP).
3. Control Line State (using NCP or CFE).
4. Monitor line status and activity (using NCP or NTD). (TRA and LBK are also used for monitoring lines. See Sections 8.2.3 and 8.2.4.)
5. Modify the physical configuration (using NCP, VNP, or CFE).

Appendixes D, E, and F describe in detail all CFE, NCP, and VNP commands that are used with the PE.

6.1 Loading and Unloading PE Software

The NCP utility can be used to load and unload the Communications Executive, and PE lines and processes, as described in the following subsections.

6.1.1 Loading the Communications Executive

To load the Communications Executive and all lines that were marked for load during PEGEN, enter the following NCP command:

```
NCP>SET SYSTEM
```

The Communications Executive remains loaded until you either unload it (see Section 6.1.2), or reboot the system.

6.1.2 Unloading the Communications Executive

To unload the Communications Executive and free all resources allocated to the system enter the following NCP command:

```
NCP) CLEAR SYSTEM
```

6.1.3 Loading a Line (and/or Changing Controller Values)

Use the SET LINE command with ALL or the loading options (see below) to load the driver process and the appropriate data bases for a line. With NCP, a line remains loaded until you either unload it with the CLEAR LINE (see Section 6.1.4) or CLEAR SYSTEM command, or reboot the system. Note that you cannot use NCP to load new lines to your PE. To do this you have to regenerate your PE system.

With the SET LINE command you can also override the following default controller values set during PEGEN, using the optional arguments described in the SET LINE command format:

- Hardware vector address
- Hardware priority
- External page address (CSR)

NOTE

You must clear all lines on a multiplexing communications device before assigning new controller lines.

The line controller values remain in effect until you:

1. Unload the line using the CLEAR LINE ALL command.
2. Reboot the system.
3. Change one or more assigned values by:
   a. Issuing a CLEAR LINE ALL command
   b. Then issuing a SET LINE command with the new controller value(s)

The format of the SET LINE command is:

SET LINE *line-id* { ALL / *loading-option(s)* / *loaded-option* }

where

line-id is the identification of the line (Section 2.2.2.3).

ALL specifies the defaults defined in the permanent database for each of the line options.

loading-option(s) consists of one or more of the options described below. To exercise any option except STATE, the line must be cleared. To set more than one option at a time, separate each option of the entered string by a space. See the examples below.

[CONTROLLER] CSR *csr-address* sets the address of the first CSR for a controller device.

[LOCATION] *location-option* controls the allocation of dynamic memory, the location-options are:

FIRSTFIT

allocates the driver process to the first available space that is large enough.

TOPDOWN

loads the driver process at the top of the partition.

PRIORITY *hardware-priority* sets the hardware priority given to the controller. *Hardware-priority* is a number (from 4-7) given a hardware device.

STATE *line-state* sets the operational state of a line. The states are:

OFF

the line is not available for any operation.

ON

the line is loaded and is in its normal operational state.

SERVICE

the line is loaded and is reserved for line service operations such as line loopback testing or downline loading a remote node.

UNIT CSR *csr-address* — sets the address of the first CSR for a device being controlled by KMC-11 (KBU).

NOTE

With the KBU, a unit CSR may be specified for any unit on the controller.

VECTOR *hardware-vector* — sets the vector address of the controller. This argument must be the hardware vector address selected on the device.

loaded-option — consists of the option below. Use this option to change the state of loaded lines.

STATE *line-state* — sets the operational state of the one of OFF, ON, or SERVICE (see above.)

NOTE

If you are using a multiplexing device, the same VECTOR *vaddr*, PRIORITY *pnum*, and CSR *pgaddr* that you first assign to any line will be assigned automatically to all lines on the device.

Example 1. To load a BU line, BU-1, with the parameters unchanged from those specified in PEGEN:

```
NCP>SET LINE BU-1 ALL
```

Example 2. To load a line with a new CSR address, 160050:

```
NCP>SET LINE BU-1 CSR 160050
```

This loads the line with the new CSR address and all other parameters unchanged.

Example 3. To change the hardware priority, in addition to changing the CSR address:

```
NCP) SET LINE BU-1 CSR 160050 PRIORITY 7
```

This simultaneously sets the CSR and priority.

6.1.4 Unloading a Line

Use the CLEAR LINE ALL command to unload a line. The line must be set to OFF before you can unload a line. (Use the STATE OFF loading option with the SET LINE command.)

The format of this command is:

CLEAR LINE *line-id* ALL

Example. To unload the line BU-1:

```
NCP) CLEAR LINE BU-1 ALL
```

6.1.5 Loading a Process

Use the SET PROCESS command to load a process. You can also use this command to override default parameters specified during PEGEN. You can set partition names, and also the maximum number of hardware controllers and lines that a process can handle.

With NCP a process remains in memory until you clear the system or the process (see 6.1.6) or reboot the system. The format of the command is:

SET PROCESS process-name $\left\{\begin{matrix}\text{ALL} \\ \text{process-option(s)}\end{matrix}\right\}$

where

process-name specifies a process. The *process-name* options are:

AUX

BSD

DLX (RSX-11M-PLUS only)

TUD

xxx (a device mnemonic: BU, KBU, DV)

ALL — loads the process with the defaults specified in the permanent database for each of the *process-options*.

process-option(s) — consists of one or more of the following:

[LOCATION] *location-option* — controls the allocation of dynamic memory. If omitted, FIRSTFIT is used. The *location-options* are as follows (you can use the first three letters of an option name, FIR and TOP):

FIRSTFIT

allocates the first available space that is large enough.

TOPDOWN

loads the driver process at the top of the partition.

MAXIMUM CONTROLLERS *count* — sets the maximum number of controllers the process can allocate space for. The *count* is a decimal number in the range 1 to 64.

MAXIMUM LINES *count* — sets the maximum number of lines the process can handle. The maximum is a decimal number in the range of 1 to 64.

PARTITION *partition-name* — sets the name of the partition to override the *partition name* from the permanent data base.

Example 1. To load the TUD process:

```
NCP)SET PROCESS TUD ALL
```

and TUD will be loaded with the default specified in the permanent database for each of the process-options.

Example 2. To change the allocation of dynamic memory so that the process is loaded at the top of the partition:

```
NCP)SET PROCESS TUD LOCATION TOP
```

6.1.6 Unloading a Process

You can unload a process only when it is not active.

Use the CLEAR PROCESS ALL command to unload a process. The format of this command is:

CLEAR PROCESS *process-name* ALL

where

process-name is one of:

AUX

TUD

BSD

DLX

xxx (a device mnemonic: BU, KBU, DV)

ALL clears all parameters associated with the process.

6.2 Start Up/Shut Down

This section discusses procedures and considerations for starting up and shutting down your 3271 PE.

6.2.1 Start Up

To start up or to restart the PE, follow the load-initialization instructions described in Section 5.1.

When starting up the PE in a running configuration where other functions are making demands upon available resources, it is important to remember that the PE's buffer pool and the following processes are not checkpointable:

CEX Communications Executive

AUX Allocation/Recovery support to CEX

BSD PE BISYNC protocol module

BU DU/DUP-11 driver

KBU KMC-11/DUP-11 driver

DV DV driver

TUD PE user program interface module

DLX Direct Access line interfaces

The buffer pool and the above processes are fixed in storage while they are active.

The following tasks are checkpointable:

NCP Network Control Program (See Appendix E for description of subset used.)

NTINIT Network Initializer (See Chapter 8 for error messages issued.)

NTL Network Loader (See Chapter 8 for error messages issued.)

CFE Configuration File Editor (See Section 6.6.3.2 and Appendix D for description of subset used.)

VNP Virtual Network Program (See Appendix F for description of subset used.)

LBK Line Loopback Test program (See Section 8.2.4.)

6.2.2 Shut Down

To bring down your PE, issue the following command for each line:

NCP) SET LINE *line-id* STA OFF

where

line-id is the identification of the line (see Section 2.2.2.3 for line specifications).

OFF indicates the line is not available for operation.

The above command turns the operational state of the PE line to OFF. (The operational state of a PE line can be either ON or OFF. When a line is in the OFF state it is not available for use.)

Note that if any TU devices are attached to the line, the PE will detach the devices and complete all I/O with an error code. All TU devices on the line will be set to offline state. (Offline state means that the line device represented by the control block is unavailable for use.) After the line is in the offline and OFF state, the line may be cleared by typing the following:

NCP) CLEAR LINE *line-id* ALL

This deallocates all resources used by the line. You must enter this command if you want to change the CSR or VECTOR address (with the SET LINE command, discussed in Section 6.1.3).

You can clear the entire PE system (including the Communications Executive, the lines, and the processes) from memory by typing:

NCP)CLEAR SYSTEM

In this case, you would not need to use the CLEAR LINE command, unless you want to change the CSR or VECTOR address for a line, as noted above.

To return to MCR type one of the following:

NCP)EXIT

or

NCP)CTRL/Z

6.3 Controlling Line State

You can control a line's operational state by using NCP or CFE. A line's state can be switched ON or OFF. When a line is ON, it is in its normal operating state. When a line is OFF, it is not available for use.

Changes made to the line state by NCP take effect immediately, while changes made by CFE will not take effect until you reboot the system or unload and reload the line.

To use NCP for controlling the line state, use the SET LINE command with the STATE *loading-option*:

NCP)SET LINE *line-id* STATE $\left\{\begin{matrix}\text{ON}\\\text{OFF}\end{matrix}\right\}$

To use CFE, see 6.6.3.2 (DEFINE LINE command).

NOTE

The synchronous clock signals should always be applied to the DUP interface when you change the state of a KBU controlled line. The KBU microcode uses the character-done transitions of the DUP to control certain operations. If clock signals are not present, the operation will not complete and the KBU microcode may suspend operations indefinitely while waiting for the DUP. In such cases, the KBU driver must be cleared from the system and reloaded. This reinitializes the KMC.

6.4 Monitoring Lines

To monitor the status of lines, use the NCP utility. To monitor the activity of lines, you can use NCP, DCE, the Trace program (TRA), or the Line Loopback Test program (LBK).

6.4.1 Monitoring Line Status

To determine the status of a line, enter the following:

NCP) SHOW LINE *line-id* STATUS

where *line-id* specifies the line whose status you wish reported.

To determine the status of all known lines, enter the following:

NCP) SHOW KNOWN LINES STATUS

An example of the display returned follows. Table 6-1 defines the status messages.

```
LINE            STATE       LOOPBACK    ADJACENT    BLOCK
                            NAME        NODE        SIZE


BU-0            ON
DV-0-0          CLEARED
KBU-0-0         CLEARED
```

The LOOPBACK NAME, ADJACENT NODE, and BLOCK SIZE columns are not used with the PE.

Table 6-1: Responses to Line Status Inquiry

Message	Meaning
CLEARED	The data bases or the processes for the line have not yet been loaded. Use SET LINE *line-id* to make the line ready.
OFF	The line is ready but it has not yet been enabled by SET STATE LINE *line-id* ON.
ON	The line is ready and enabled.
Invalid line ID	The specified *line-id* does not exist in the active configuration. This is an error message which is returned only when using the SHOW LINE *line-id* specification.

6.4.2 Monitoring Line Activity

The NCP and DCE utilities monitor line activity. These utilities have similar features, but DCE also maintains log files of events on a time-initiated basis. With NCP you can zero line counters.

TRA can be used as a line monitor but is more useful as a debugging aid (see Chapter 7).

6.4.2.1 Using NCP to Monitor Line Activity — With NCP's SHOW LINE COUNTERS command, the 3271 PE provides a special display that is modified to suit the PE environment. Eight counters monitor each line and record the occurrence of specific types of activity. To get a reading of a set of counters for a particular line, enter the following:

NCP) SHOW LINE *line-id* COUNTERS

A sample display that might then be generated follows:

```
Line counters as of dd-mon-yy hh:mm

Line = BU-0

       10  Seconds since last zeroed
      963  Data blocks received
      412  Data blocks sent
        0  NAKs transmitted
        0  NAKs received
        0  Acknowledgment timeouts
        0  WACKs transmitted
        0  NAK thresholds exceeded
        0  ENQ thresholds exceeded
        0  ACKs received out of sequence
        0  3271 command rejects
        0  Transmission aborted by the host
        0  Receptions aborted by the host
        0  Transmit clock losses
        0  Receive clock losses
        0  Hardware errors
```

NOTE

Table 6-2 defines each activity type listed above. Chapter 8 tells how counters may be used in evaluating line and physical link performance.

To get readings for all known lines, enter the following:

NCP)SHOW KNOWN LINES COUNTERS

If you want to reset your PE's counter values to zero, enter the following:

NCP) ZERO LINE *line-id*

The command

NCP) ZERO KNOWN LINES COUNTERS

will reset counters for all known lines.

6.4.2.2 Using DCE to Monitor Line Activity — The error logging and statistics routine (DCE) provides statistics and reports on all active lines that use the BSD protocol module.

Generating the Error Counter Report. There are three ways to invoke the line monitor task to generate the error counter report (the Protocol Error Summary):

1. MCR) DCE

2. MCR) RUN [*X*,54]DCE

3. MCR) RUN ...DCE 1S/RSI=10M

where

X is the User Identification Code (UIC) group number that was specified for the 3271 Protocol Emulator (PE) to be built in.

If the line is not active and therefore DCE cannot monitor, an error message will be returned to the terminal.

If either of the first two options are used the report is generated only once, immediately after execution of the command.

You have to be a privileged user to use the third form. DCE generates the report automatically and gives you standard RSX RUN command timing options (1S/RSI=10M). You can specify:

- **When DCE report generation is to begin.** Indicate the time as an increment from the time of issue of the command. In the example above, 1S signifies that the report is to be generated one second after execution of the command.

- **The report reschedule interval.** For instance, RSI=10M means DCE is to run at 10-minute intervals.

To stop DCE when using this third method enter:

MCR) CAN ...DCE

The report is not output to a terminal but generated as a disk file with ASA FORTRAN carriage control. You can either spool it by means of the Peripheral Interchange Program or print it directly using a line printer or other terminal. The filename of the report is SY:[target UIC]TULOG.TXT, unless specified otherwise during PEGEN.

Restarting Error Report Collection. Each time you run the DCE program, it appends new data to the existing log file. If you want to restart data collection in a new file, delete each previous data file after output. Select the log device, UIC, and file name at time of PE generation (see the Components Parameters section of PEGEN dialog).

Types of Statistics and Errors Reported. Figure 6-1 shows the format of the error counter report. The counters contain decimal values (five digits maximum). An explanation of each line of the report is given in Table 6-2.

```
PROTOCOL ERROR SUMMARY                dd-mon-yy hh:mm:ss

STATISTIC ACCUMULATED

LINE 0 HALF DUPLEX DU?-11 BISYNC
LINE STATE:          HOST IS       ACTIVE
TOTAL NUMBER OF DATA BLOCKS TRANSMITTED 15893
TOTAL NUMBER OF DATA BLOCKS RECEIVED 0
TOTAL NUMBER OF NAKS TRANSMITTED 0
TOTAL NUMBER OF NAKS RECEIVED 52
TOTAL NUMBER OF ACKNOWLEDGEMENT TIMEOUTS 6
TOTAL NUMBER OF WACKS TRANSMITTED 0
TOTAL NUMBER OF NAK THRESHOLDS EXCEEDED 0
TOTAL NUMBER OF ACK TIMEOUT THRESHOLDS EXCEEDED 1
TOTAL NUMBER OF 3270 COMMAND REJECTS 0
TOTAL NUMBER OF ACKS RECEIVED OUT OF SEQUENCE 0
TOTAL NUMBER OF XMITS ABORTED BY HOST 0
TOTAL NUMBER OF RECEPTIONS ABORTED BY HOST 0
TOTAL NUMBER OF TRANSMIT CLOCK LOSSES 0
TOTAL NUMBER OF RECEIVE CLOCK LOSSES 0
TOTAL NUMBER OF HARDWARE ERRORS 7
```

Figure 6-1: DCE Error Counter Report

Table 6-2: 3271 PE Error Counters

LINE 0 HALF DUPLEX (DU?11) BISYNC (DV11)

Explanation: This line identifies the 3271 PE line number and lists its characteristics. This report and those that follow are repeated for each 3271 PE line specified during PEGEN.

LINE STATE: HOST IS ACTIVE

Explanation: HOST ACTIVE means that a poll was received from the host within the last 5 minutes. If a poll was not received in that time, the line state indicates HOST INACTIVE.

TOTAL NUMBER OF DATA BLOCKS TRANSMITTED

Explanation: Each time a data block is sent, the appropriate counter is incremented.

TOTAL NUMBER OF DATA BLOCKS RECEIVED

Explanation: Each time a data block is received, the appropriate counter is incremented.

TOTAL NUMBER OF NAKS TRANSMITTED

Explanation: Each time the 3271 sends a NAK (du to a CRC error on a received block), the appropriate counter is incremented.

TOTAL NUMBER OF NAKS RECEIVED

Explanation: Each time the 3271 receives a NAK from the host, the appropriate counter is incremented.

TOTAL NUMBER OF ACKNOWLEDGMENT TIMEOUTS

Explanation: This counter records the number of times the 3271 PE failed to receive an acknowledgment to a transmit block within 3 seconds of the time the last character was sent on the line.

TOTAL NUMBER OF WACKS TRANSMITTED

Explanation: This counter records the number of times a block was received with no user-assigned buffer available.

TOTAL NUMBER OF NAK THRESHOLDS EXCEEDED

Explanation: This counter records the number of times too many NAKs to the same block were received. This threshold value is specified during system generation as NAK COUNTER (default = 10 octal).

Table 6-2 (Cont.): 3271 PE Error Counters

TOTAL NUMBER OF ACK TIMEOUT THRESHOLDS EXCEEDED

Explanation: This counter records the number of times too many successive 3-second ACK timeouts occurred with no response from the host. This threshold value is specified during system generation as RETRY COUNTER (default = 10 octal).

TOTAL NUMBER OF 3270 COMMAND REJECTS

Explanation: This counter records the number of illegal commands received from the host (such as COPY). The 3271 Protocol Emulator Programmer's Guide describes the 3270 commands that may be used with 3271 PE.

TOTAL NUMBER OF ACKS RECEIVED OUT OF SEQUENCE

Explanation: This counter records the number of times the wrong ACK was received from the host in response to a transmitted block (such as an ACK 0 response to the first data block).

TOTAL NUMBER OF XMITS ABORTED BY HOST

Explanation: This counter records the number of times the host terminated a transmission by the 3271 PE before it acknowledged the last block (such as an EOT response instead of an ACK response to a data block).

TOTAL NUMBER OF RECEPTIONS ABORTED BY HOST

Explanation: This counter records the number of times the host sent an EOT block before sending an ETX block.

TOTAL NUMBER OF TRANSMIT CLOCK LOSSES

Explanation: Whenever a message takes more than 3 seconds to send, the appropriate counter is incremented.

TOTAL NUMBER OF RECEIVE CLOCK LOSSES

Explanation: Whenever a message takes more than 3 seconds to receive, the appropriate counter is incremented.

TOTAL NUMBER OF HARDWARE ERRORS

Explanation: Hardware errors do not necessarily involve a fault in the equipment, but rather errors that are detected by the hardware. The most common errors reflected in this count are receive overruns and transmit underruns. These are caused when the system runs at a priority higher than the device priority for more than one character time. True errors, such as NXM or memory parity errors, may also be included in this count.

6.5 Monitoring the System with NCP

PE users can show system information and monitor allocation of communications buffers by using the NCP SHOW SYSTEM command. This tool can help you tune the system.

Use the SHOW SYSTEM command to display the system information. You can direct the output of the SHOW SYSTEM command to the calling terminal or to any local output file or device.

The format of the command is:

SHOW SYSTEM [*display-type*] [TO *output-file*]

where

display type is one of the following:

CHARACTERISTICS	displays static parameters associated with the system.
COUNTERS	displays statistics or error counters associated with the system.
STATUS	displays dynamic values associated with the system.
SUMMARY	is the default *display-type*. It displays only the most useful characteristics and dynamic values associated with the system. Examples of the *display-types* follow.
TO *output-file*	directs the output to the specified file or device. The data is transferred in ASCII in the same format as it is displayed at the terminal. If omitted, the information is printed at the calling terminal. If the specified output file already exists, the information will be appended.

Example 1. To display the contents of the system counters, use this command:

NCP) SHOW SYSTEM COUNTERS

An example of the disp ay is:

```
SYSTEM COUNTERS AS OF dd-mon-yy hh:mm

12      SECONDS SINCE LAST ZEROED
 0      CONTROL BUFFER ALLOCATION FAILED
 0      SMALL BUFFER ALLOCATION FAILED
 0      LARGE BUFFER ALLOCATION FAILED
 1      RECEIVE BUFFER ALLOCATION FAILED
```

Example 2. To display system characteristics, use this command:

NCP) SHOW SYSTEM CHARACTERISTICS

An example of the display is:

```
SYSTEM CHARACTERISTICS AS OF dd-mon-yy hh:mm

MAXIMUM CONTROL BUFFERS = 200, MAXIMUM SMALL BUFFERS = 0
MAXIMUM LARGE BUFFERS = 200, LARGE BUFFER SIZE = 280
MAXIMUM RECEIVE BUFFERS = 199
```

Example 3. To display the system status, use this command:

NCP) SHOW SYSTEM STATUS

An example of the display is:

```
SYSTEM STATUS AS OF dd-mon-yy hh:mm

ACTIVE CONTROL BUFFERS = 41, ACTIVE SMALL BUFFERS = 0
ACTIVE LARGE BUFFERS = 20,
```

Example 4. To display a system summary, use this command:

NCP) SHOW SYSTEM SUMMARY

An example of the display is:

```
SYSTEM SUMMARY AS OF dd-mon-yy hh:mm

ACTIVE CONTROL BUFFERS = 20, ACTIVE SMALL BUFFERS = 0
ACTIVE LARGE BUFFERS = 12, MAXIMUM CONTROL BUFFERS = 150,
MAXIMUM SMALL BUFFERS = 0
MAXIMUM LARGE BUFFERS = 120
```

6.6 Modifying the Configuration

This section explains how you can modify your PE configuration. You can make most modifications without having to regenerate your PE by using the NCP, VNP, or CFE utilities.

NOTE

In the following situations, however, you may have to regenerate your PE system:

- When adding processes
- When changing the number of communications lines
- When changing the number of TU devices
- When changing communications controllers (for example, switching from a BU to a KBU)

6.6.1 Reloading/Renaming Process Partitions

If you want to rename or reload processes, you must enter the following NCP instructions in the specified sequence:

Step 1. NCP) SET LINE *line-id* STATE OFF

Step 2. NCP) CLEAR PROCESS *process*

Step 3. NCP) SET PROCESS *process* PARTITION *partnm*

where

line-id specifies any line associated with the process

process is any of the following:

AUX
TUD
BSD
BU
DV
KBU
DLX (RSX-11M-PLUS systems only)

partnm is the name of the partition into which the specified process will be loaded. This *partnm* will override the partition name specified for the process in PEGEN.

6.6.2 Adding Lines

As mentioned previously, to add lines to the configuration, you must regenerate the PE. Re-execute the PEGEN process described in Chapter 4.

6.6.3 Changing Configuration Parameters

You can change parameters you established in PEGEN by:

- Modifying the currently running system, using NCP
- Modifying the system image file, using VNP
- Modifying the configuration file, CETAB.MAC, using CFE

NCP changes take effect immediately but apply only to the running system.

When you use VNP, the changes take effect only when the system is rebooted. (VNP is not supported for RSX-11M-PLUS systems.)

When you use CFE, the changes do not take effect until you reboot the system or unload and reload the affected part of the software. If you have already used VNP to load PE system lines or processes into the system image file, CFE line and process changes do not come into effect, unless you first clear the lines and processes using VNP. Always display the system with VNP before making CFE changes by typing:

```
VNP)SHOW SYSTEM SUMMARY
```

CFE affects some parameters that are inaccessible to NCP and VNP, such as Communications Executive parameters.

You can change PE controller (line interface) and station addresses originally specified during PEGEN by modifying the PE.MAC file, using any text editor. You must then reassemble and build the BSD process. Section 6.6.4 describes this procedure.

6.6.3.1 Modifying a Generated Configuration Using NCP or VNP — Using NCP or VNP, you can set and clear information for the communications lines and Communication Executive processes. The information includes the following parameters and states of your PE system:

- Line controller and unit CSR address, priority, state, and vector
- Process location in memory, maximum controller count, maximum lines count, and partition name

With NCP or VNP, modify line parameters using the SET LINE command with the loading-options. When modifying parameters other than the line STATE, you must first clear the lines (using the CLEAR LINE command). Sections 6.1.3 and 6.1.4 discuss the use of these commands. To modify process parameters, use the SET PROCESS command, (see Section 6.1.5).

Once parameters are set, they remain set until another NCP or VNP command resets or clears them, or until the system is rebooted.

Appendixes E and F describe the NCP and VNP SET and CLEAR commands in detail.

6.6.3.2 Modifying the Configuration Using CFE — The Configuration File Editor Utility (CFE) is used to change or to display the parameters of the configuration file, CETAB.MAC, that was created during PEGEN. CFE commands do not affect the running system. Changes take effect when the system is rebooted. Using CFE, you can redefine the PE system parameters listed in Table 6-3:

Table 6-3: CETAB Values Changed by CFE

CETAB Parameter	What You Can Change with CFE
Buffer Definitions	Number of Communications Control Blocks (CCBs) Number of Receive Data Buffers (RDBs) RDB threshold value Number of transmit buffers
Network Pool Parameters	Dynamic allocation Partition names Top-down loading
Process Definitions	**For each process selected during PEGEN:** Mark the process for load
Device Definitions	**For each controller on each device:** Vector address CSR address Controller priority
	For each line on each controller: Secondary CSR address (For KMC11 devices only)
Line Status	**For each system line:** Mark the line for load Enable the line

Invoking CFE

Before invoking CFE, you must know the device and UIC under which the configuration file resides. Following PEGEN, CETAB.MAC resides under the UIC:

[*x*,54]

where

x is the UIC selected during PEGEN.

Invoke CFE by typing:

Method 1 MCR) CFE

CFE then returns the prompt:

```
ENTER FILENAME:
```

Enter the name of the configuration file that you want to change. If you omit any or all of the filename specifications, the default is SY: [current UIC] CETAB.MAC.

Enter the CFE command immediately following the prompt.

Method 2 MCR) CFE *filename*

where

filename is the name of the configuration file that you want to change.

CFE displays the CFE prompt. Enter the CFE command immediately following the prompt.

Method 3 MCR) CFE *@command file* [/TR]

where

command file is the name of a file containing CFE commands and [/TR] is an optional switch that allows you to trace the commands as they are executed.

The first entry in the command file must contain the name of the configuration file you want to change. The remainder of the file should contain one or more CFE commands. You can nest command files.

Exiting CFE

Use the EXIT command or a CTRL/Z to terminate a CFE session.

Format: EXIT or CTRL/Z

Use the EXIT PURGE command to terminate an editing session, update the configuration file, and delete the old file.

Format: EXIT PURGE

You can use the KILL command to abort an editing session, but execution of this command will not cause the configuration file to be updated, and will not delete the old file.

Format: KILL

Using CFE

With CFE you can modify PE component parameters by using the DEFINE command. You can display parameters of components by using the LIST command. You may wish to use the LIST command to determine the parameters set during PEGEN, or to verify that the correct changes were made after modifying parameters with the DEFINE command.

The rest of this section tells you how to use CFE to do the following:

- Display line characteristics
- Modify line parameters
- Display process characteristics
- Modify process parameters
- Display system characteristics
- Modify system parameters

Appendix D is a reference for all CFE commands that can be used with the PE. It describes command syntax, and illustrates a sample CFE session.

Displaying Characteristics of Lines

You can display the characteristics of a specific line by typing:

LIST LINE *line-id*

If, for example, you want to display the characteristics of all known lines, type:

LIST KNOWN LINES

If, for example, you want to display the parameter settings for line DV-0-0, type:

LIST LINE DV-0-0

An example of the display returned is:

```
Line characteristics as of dd-mon-yy

Line = DV-0-0
  Controller CSR = 17500, Vector = 340, Priority = 5
  Duplex = Half, Speed = Sy" hronous
  Cost = 0, State = Off
```

Modifying Line Parameters

Use the DEFINE LINE command to change parameters set during PEGEN. You can set the parameters for a specific line by typing:

DEFINE LINE *line-id line-option(s)*

To set parameters for all known lines, type:

DEFINE KNOWN LINES *line-option(s)*

The *line-options* are:

[CONTROLLER] *csr-address*	CSR sets the address of the first CSR for the line controller.
PRIORITY *priority*	indicates the hardware priority of the controller. This argument is a number from 4 to 7, given to a hardware device.
STATE *line-state*	sets the line's operational state. The possible states are as follows:

CLEARED

The line is not loaded

OFF

The line is loaded and not being used.

STATE	ON The line is available for normal use with the exception of temporary overrides for service functions.
UNIT CSR *csr-address*	indicates the address of the first CSR for a device being controlled by a KMC-11.
VECTOR *vector-address*	indicates the vector address of the controller. This argument must be the hardware vector address selected on the device.

You can specify more than one option in a single DEFINE LINE command. Separate each option with a single space.

You can modify parameters for all known lines by typing:

```
DEFINE KNOWN LINES
```

Example. To change the hardware priority and state of BU-0, type:

```
DEFINE LINE BU-0 PRIORITY 5 STATE OFF
```

Displaying Characteristics of Processes

Use the LIST PROCESS command to display the characteristics of a process. To display information about a specific process, type:

```
LIST PROCESS process-name
```

For information on all known processes, type:

```
LIST KNOWN PROCESSES
```

Process-names include BSD, TUD, AUX, DLX, or a device mnemonic (BU, KBU, or DV).

Example. If you want to display characteristics of the BSD process, type:

```
LIST PROCESS BSD
```

An example of the display returned is:

```
Process characteristics as of dd-mon-yy hh:mm:ss

Process = BSD
  State = Cleared
  Maximum lines = 16, Maximum controllers = 2
```

Modifying Process Parameters

Use the DEFINE PROCESS command to change the load parameters for a specific process or all known processes. For a specific process, type:

DEFINE PROCESS *process-name process-option(s)*

To modify all known processes, type:

DEFINE KNOWN PROCESSES *process-option(s)*

The process options are:

MAXIMUM CONTROLLERS *count* — sets the maximum number of hardware controllers for which the process is to allocate data space. The number is decimal in the range 1 to 64.

MAXIMUM LINES *number* — sets the maximum number of lines for which the process is to allocate data space. The number is decimal in the range 1 to 64

STATE *process-state* — sets the process's operational state. The possible states are as follows:

CLEARED

The process is not available for use and must be manually loaded.

ON

The process is available for normal use and is automatically loaded.

OFF

The process is loaded but not available for use.

You can specify more than one option in a single DEFINE PROCESS command. Separate each option with a single space.

Example. To set the maximum number of lines for which a DV is to allocate data space, type:

```
DEFINE PROCESS DV MAXIMUM LINES 4
```

Displaying System Characteristics

To display system characteristics, such as buffer parameters and partition information, type:

LIST SYSTEM

Example. An example of a display in response to the LIST SYSTEM command is:

```
System characteristics as of dd-mon-yy

Pool name = POOL...Partition name = GEN
Block-area extension = 53, Byte-area extension = 64
Location = Topdown

Maximum control buffers = 20, Maximum small buffers = 0

Maximum large buffers = 35, Large buffer size = 290
Receive buffer minimum = 3
```

Modifying System Parameters

The DEFINE SYSTEM command changes the Communications Executive system parameters, such as large buffer size, maximum number of control buffers and large buffers, and pool partition name (see the LIST SYSTEM display example above for all parameters that can be affected). To change system parameters, type:

DEFINE SYSTEM *system-option(s)*

where

the system options consists of one or more of the following:

LARGE BUFFER — sets the size (in bytes) of the large buffers.

SIZE *number* — The SIZE is specified as a decimal number in the range 256 to 4096.

[LOCATION] *location-option* — controls the allocation of dynamic memory.

NOTE

This option is allowed only if the network pool is dynamically allocated (specified during PEGEN).

where

FIRSTFIT

specifies that processes will be loaded in the first available space that is big enough.

TOPDOWN

specifies that processes will be loaded at the top of the partition.

MAXIMUM CONTROL BUFFERS *number*	sets the number of control buffers available for Communications Executive system use. The number is a decimal value in the range 1 to 200.
MAXIMUM LARGE BUFFERS *number value*	sets the maximum number of large buffers available for communications. The number is a decimal in the range 1 to 200.
MINIMUM RECEIVE BUFFERS *number*	sets the minimum number of receive buffers available for communications. The number is a decimal value in the range 1- (maximum large buffers -1).
POOL BLOCK AREA *block-count*	is the number of blocks needed for the network pool. This is a decimal number in the range 0 to 128.
POOL BYTE-AREA *byte-count*	is the number of bytes needed in the network pool. This is a decimal number in the range 0 to 8192.
POOL PARTITION *partition-name*	indicates the name of the partition into which the network pool is loaded.

6.6.4 Modifying PE Controller and Station Addresses

During PEGEN you specified the PE control unit and station addresses to be recognized by the PE lines during polling from the host system (see Section 3.5 for an example illustrating how these addresses are assigned). These addresses are assembled and built into the BSD process during the build phase at the end of the PEGEN dialog. If you want to change these addresses later, you may do so either by doing another complete PEGEN or by modifying the current addresses contained in the PE.MAC file and then reassembling and building the BSD process.

To modify these addresses using the latter method, perform the following steps:

Step 1. Use any editor to modify the controller and station addresses in the PE.MAC file. The complete filename for this file is:

|*g*,10|PE.MAC

where

g is the group number under which all generated files reside (check your response to Question 6 of the Preliminary section of PEGEN).

Use the editor to search for the symbol definitions for the controller and station addresses to be modified. The symbol definitions are:

B$$*An*=STA*xx* for controllers
An$*m*=STA*xx* for stations

where

n is the controller number
m is the station number
xx is the address number.

Step 2. Change the STA*xx* to the desired address using Table 3-3 (Chapter 3). For example, if BU-0 was generated for controller address STA1 and station addresses STA1 and STA2, and you want it to have controller address STA0 and station addresses STA0 and STA1, then you would modify the PE.MAC file as follows:

Before:

```
B$$A0=STA1      ;CU 0 Address
;               ;CU 0 Trace Unit,      TU 0
A0$1=STA1       ;CU 0 DEV 1 Address,   TU 1
A0$2=STA2       ;CU 0 DEV 2 Address,   TU 2
```

After:

```
B$$A0=STA0      ;CU 0 Address
;               ;CU 0 Trace Unit,      TU 0
A0$1=STA0       ;CU 0 DEV 1 Address,   TU 1
A0$2=STA1       ;CU 0 DEV 2 Address,   TU 2
```

Step 3. After modifying the PE.MAC file, reassemble and build the BSD process as follows:

NOTE

The BSDBLD.CMD file will assume the same disk assignments for the target and distribution disks as were specified during PEGEN. Make sure that you first verify that the disks are assigned and mounted correctly.

1. Assign the name of your target disk to the default device, SY:. For example, if your target disk is DM1:, type:

```
) ASN SY:=DM1:
```

2. Assign the name of your distribution disk to the logical name DI:. For example, if your distribution disk is DM0:, type:

```
)ASN DI:=DM0:
```

3. Set the default UIC for the entering terminal as [*g*,24] where *g* is the group number of the UIC under which PE.MAC and all other generated files reside. For example, if the group code is 377, type:

```
)SET /UIC=[377,24]
```

4. Use the MACRO-11 assembler to assemble an object file and listing file named BSDX. The source files are PE.MAC and BSDX.MAC, and the library file NETLIB.MAC. Issue the MACRO-11 command as follows, where *g* is the group number of the UIC under which all generated files reside:

```
)MAC BSDX,g,34=BSDX/-SP=
DI:[130,10]NETLIB/ML,SY:[g,10]PE,BSDX
```

5. Taskbuild the files using the indirect command file BSDBLD:

```
)TKB @BSDBLD
```

Step 4. When finished, you will have a new version of BSD with the modified addresses. These addresses take effect the next time BSD is loaded. If the PE is currently running, you must turn all lines off, clear them, and then load and start them again.

6.6.5 Reconfiguring with RSX-11M-PLUS

The RSX-11/3271 PE supports the fault isolation and system reconfiguration features of the RSX-11M-PLUS operating system. The reconfiguration task CON may be used to place a PE device off line and later return it to an on-line state. Note that CON operates on controllers and that the effect of taking a PE controller off line may cause several lines to be turned off.

The syntax used by CON is described in the *RSX-11M-PLUS Fault Isolation Guide*. However, the devices and controllers are named using the RSX name convention rather than the PE name convention. Table 6-4 associates the RSX-11M-PLUS names with the PE name.

Table 6-4: RSX-11M-PLUS Device Names

PE Name	RSX-11M-PLUS Name
BU	WB
KBU	WQ
DV	WV

The first controller of a given type has the letter A appended, the second controller B, etc.

Examples:

BU-0 = WBA
KBU-0-* = WQA
DV-1-* = WVB

Chapter 7
3271 Protocol Emulation, Message Formats, and Operation Sequences

The RSX-11/3271 Protocol Emulator uses a subset of IBM's Binary Synchronous Communications Protocol (BSC, also called BISYNC). The BISYNC protocol includes a set of data-link control characters that are inserted as part of the information stream transmitted across a communications line. These control characters regulate transmission on a synchronous, serial communications facility.

This chapter discusses the 3271 PE communications protocol, BISYNC data-link characters, message formats, and operation sequences for polling, READ, selection, and WRITE operations.

7.1 Overview of 3271 Protocol Emulation and Communications

The 3271 PE responds when the host system polls or selects a PE station.

The host polls all stations to determine their current status and whether they are in an I/O pending condition. When the host detects an I/O pending condition at a PE station (for example, when the station has a message to transmit), polling stops and the host communicates solely with that PE station. The PE station then transmits its message, inserting certain control characters and other codes in the message stream. These instruct the host system to perform a READ operation. The host responds to the READ operation by storing the data message in its main memory for subsequent analysis by the host's access method. When communication ends, the host system begins polling at the next sequential station.

The host selects a particular PE station to which it wishes to transmit data. When selected, the PE station receives data from the host only upon acknowledging (by sending certain data-link control characters) that it is ready to receive. The host system initiates data transmission to a PE station with a WRITE command.

Each PE station is assigned a unique address that is recognized by the host system during either polling or selection. Tables 7-4 and 7-6 list the valid polling and selection addresses for the 3271 PE.

The 3271 transmission code set has four types of characters:

1. Graphic characters
2. End-to-end characters
3. Data-link control characters
4. Block Check characters (BCC)

Graphic characters are the alphanumeric characters and other characters, such as period, comma, and dollar sign.

End-to-end characters are all address, command, and order characters sent to the 3271 PE, and all AID bytes, and status/sense bytes read from the 3271 PE. The end-to-end control characters are used by the terminal station for data control and data format functions. Different terminal devices require different end-to-end control functions. For example, a display may require a cursor control character which would mean nothing to a line printer. The way in which end-to-end characters are interpreted and handled depends on the RSX user's application program. End-to-end characters are described in detail in the *RSX-11/3271 Protocol Emulator Programmer's Guide*.

Data-link control characters and their functions are described in Section 7.2. The 3271 PE adds data-link control characters to messages transmitted to the host system. The PE strips away control characters from messages received from the host before passing the data to an RSX user application program. (See Figure 7-1.)

Block Check characters are generated by the 3271 PE to check for accuracy of all transmissions. The PE performs calculations (called Cyclic Redundancy Checking) based on the numeric binary values of each block of a message, and generates, for each block, a 2-byte BCC character. When transmitting, the PE appends a BCC to each block of data. When receiving a block of data, the PE compares the received BCC with its own computed BCC. If they are equal, the PE accepts the message. If unequal, the PE requests the host to retransmit the message.

7.2 BISYNC Data-Link Control Characters

BISYNC is a half-duplex (transmit/response) protocol requiring synchronization and a response for each message block and each control sequence. BISYNC data-link characters are used by the BISYNC protocol to regulate transmission, facilitate error detection, and provide station synchronization. The characters defined in the data-link control character subset used by the 3271 PE are:

ACK 0/ACK 1: Even/Odd Positive Acknowledgement

DLE: Data Link Escape

ENQ: Enquiry

EOT: End of Transmission

ETB: End of Transmission Block

ETX: End of Text

NAK: Negative Acknowledgement

RVI: Reverse Interrupt

SOH: Start of Header

STX: Start of Text

SYN: Synchronous Idle

TTD: Temporary Text Delay

WACK: Wait Before Transmit Positive Acknowledgement

These control characters are represented as either a single character 8-bit sequence or a 2-character 16-bit sequence. The characters are used in the message formats and operation sequences that are described later in this chapter. Each of the data-link control characters is discussed in the following sections, in context of three functional groupings:

1. BISYNC Equipment Control characters
2. Message Framing Control characters
3. Message Interchange Control characters

7.2.1 Equipment Control Characters

The equipment control characters are used to ensure proper functioning and synchronization of the data link. They include the following two control characters:

SYN: Synchronous Idle

DLE: Data Link Escape

SYN — Synchronous Idle. The Synchronous Idle character helps synchronize the sending and receiving stations. The 3271 PE may use from 3 to 15 consecutive SYN characters to establish synchronization. In PEGEN you establish the number of SYN characters to be used. SYN characters may also be embedded in the transmitted text by the host for time-fill to maintain synchronization (the 3271 PE does not store such SYN characters). The SYN characters are deleted from the message by the receiving station and are not included in the block check count (BCC) accumulation at either station.

DLE — Data Link Escape. The DLE character redefines the meaning of the character immediately following. It is used by the 3271 PE to provide supplementary line control characters (WACK, ACK 0/ACK 1, and RVI). The DLE appears as the first byte, followed by the supplementary control character.

Table 7-1, following, summarizes the BISYNC Equipment Control characters, giving their hexadecimal representations and their various functions.

Table 7-1: Functions and Hexadecimal Representation of BISYNC Equipment Control Characters

Data-Link Control Characters	Hexadecimal Bit Configurations	Function 1	Function 2
SYN Synchronous idle	32	Causes initial transmitter/receiver synchronization.	Maintains synchronization between transmitter and receiver.
DLE Data Link Escape	10	Identifies two-character control sequence.	

7.2.2 Message Framing Control Characters

These control characters are used to frame a message or a portion of a message. In other words, they indicate the beginning and end of a message or message block. They activate the hardware checking circuitry at the transmitting and receiving stations. The Message Framing Control characters consist of the following:

SOH: Start of Header

STX: Start of Text

ETB: End of Transmission Block

ETX: End of Text

SOH — Start of Header. SOH precedes and identifies the header portion of a message. The 3271 PE generates SOH in a three-character heading sequence to identify the data as either a STATUS or TEST REQUEST message:

STATUS message	SOH % R STX...
TEST REQUEST message	SOH % / STX data...

Section 7.3.3 discusses STATUS message format. TEST REQUEST messages include user-generated data. The format of these messages is distinguished from that of a READ MODIFIED message by the SOH % / header.

Upon receipt of the SOH, BCC accumulation begins. The SOH is not included in the accumulation. Subsequent STX characters are considered to be data characters and are included in the BCC.

STX — Start of Text. STX precedes and identifies a block of normal text data or a TTD sequence. The 3271 PE transmits STX as the first character of a read-data block, except in a STATUS or TEST-REQUEST message.

The first character in STATUS and TEST-REQUEST messages is SOH, with STX following the two header characters. With this type of message the BCC accumulation starts on receiving the SOH. The following STX character is included in the BCC. The first SOH/STX sequence in a message block that is not preceded by a header also causes BCC accumulation to begin, but the STX is not included in the accumulation.

ETB — End of Transmission Block. ETB signals the end of a block of data and informs the receiving unit that the BCC is the next character and that other block(s) will follow. It causes the receiver to turn the line around and transmit the appropriate response. Contrast this character with the ETX character described below.

ETX — End of Text. ETX indicates the end of the last (or only) block of text and informs the receiving unit that the BCC follows. ETX causes the receiver to turn the line around and transmit the appropriate response. Note that the ETX indicates the last block transmitted, while the ETB character indicates the end of a block of data, but not the last block transmitted.

Table 7-2 summarizes the Message Framing Control characters, giving their hexadecimal representation and their various functions.

Table 7-2: Functions and Hexadecimal Representation of BISYNC Message Framing Control Characters

Data-Link Control Characters	Hexadecimal Bit Configurations	Function 1	Function 2	Notes
SOH Start of Header	01	Indicates a message block containing header information.	Activates the message checking accumulators, if first SOH/STX in message block.	
STX Start of Text	02	Indicates a message block containing text.		
ETB End of Transmission Block	26	Indicates the end of a message block with more blocks to follow.	Stops the message checking accumulators.	Requires a response from the receiving station.
ETX End of Text	03	Indicates the end of the last block in a message.	Stops the message checking accumulators	

7.2.3 Message Interchange Control Characters

These control characters affect the line activity directly. They are:

EOT: End of Transmission

ENQ: Enquiry

NAK: Negative Acknowledgement

and the following two-character sequences:

ACK 0(DLE"70")/ACK 1 (DLE"61"): Positive Acknowledgement

WACK(DLE"6B"): Wait before Transmit Positive Acknowledgement

TTD(STX ENQ): Temporary Text Delay

RVI(DLE"7C"): Reverse Interrupt

7.2.3.1 Single Characters

EOT — End of Transmission. The EOT indicates end of transmission from either the host or the 3271 PE. In either condition, line synchronization is abandoned and the 3271 PE returns to control mode (waiting to be polled or selected). The EOT may also be used as:

- A response to a poll when there is no data to transmit or no pending status
- A response to a selection when the PE is unable to perform an operation
- An answer to a Reverse Interrupt (RVI) issued by the host system

ENQ — Enquiry. The ENQ character is used:

- To request a reply from the host system following a three second timeout
- To request retransmission of the previous reply from the host system
- As part of a Temporary Text Delay (TTD) message, to signal a transmitter abort sequence

When the PE receives an ENQ in response to a transmission, it repeats the last transmission to the host. The PE responds with NAK when ENQ is:

- Received as the last character of a text message aborted by the host system.
- Embedded in text.
- Received as part of a TTD sequence.

ENQ is also received by the PE as the last character of a poll or selection sequence on a non-switched data-link, or as a bid by the host system on a point-to-point data-link.

NAK — Negative Acknowledgement. A receiving station transmits a NAK character to indicate reception of a faulty text block or of a TTD sequence. For the former, the 3271 PE retransmits the last block of text. In response to a TTD sequence, the 3271 PE will send an EOT.

A NAK may also be transmitted in response to a host system transmission block that:

- Is missing an ETB or ETX character
- Has an ENQ embedded in text.

7.2.3.2 Two-Character Control Sequences

ACK 0(DLE 70) — Even Positive Acknowledgement. ACK 1(DLE 61) — Odd Positive Acknowledgement. These character responses, in the proper alternating sequence, indicate that the previous block was accepted without error and the receiver is ready to accept the next block of the message. ACK 0 is the positive reponse to:

- A selection sequence (non-switched, point-to-point data link)
- A line bid (switched data link)
- An even-numbered text block

ACK 1 Is the positive response to an odd-numbered text block.

ACK 1 acknowledges receipt of the first and all odd-numbered data blocks, and ACK 0 acknowledges receipt of the second and all even-numbered data blocks. This alternating response provides a sequential control check to determine if there have been any lost blocks of text.

WACK(DLE 6B) — Wait before Transmit Positive Acknowledgement. WACK indicates a temporary not-ready-to-receive condition. The 3271 PE transmits a WACK to acknowledge that it has received a block of text but there are no internal buffers left. If the PE receives a WACK while transmitting, it sends an ENQ. The PE also transmits a WACK for any TU unit attached as a printer, when the last block (ETX) of a message is received.

TTD(STX ENQ) — Temporary Text Delay. The 3271 PE uses this sequence to inform the host system that it is aborting the current transmission. This is known as a forward abort sequence. After receiving a NAK response to this TTD, the PE sends an EOT, resetting both stations.

The PE may also use this sequence to retain the line when it is not ready to transmit.

If the PE receives a TTD sequence from the host, it responds with a NAK.

RVI(DLE 7C) — Reverse Interrupt. The 3271 PE sends an RVI in response to an attempted selection sequence by the host system when the PE has a status message (other than device busy or device end) to be transmitted. The host system yields control by sending an EOT.

Whenever the PE receives an RVI from the host system, it responds with EOT and resets all pending status and sense information. (See Section 7.3.3 for a discussion of status and sense information.)

Table 7-3 summarizes the Message Interchange Control characters, their hexadecimal representation, and their various functions.

Table 7-3: Functions and Hexadecimal Representations of BISYNC Message Interchange Control Characters

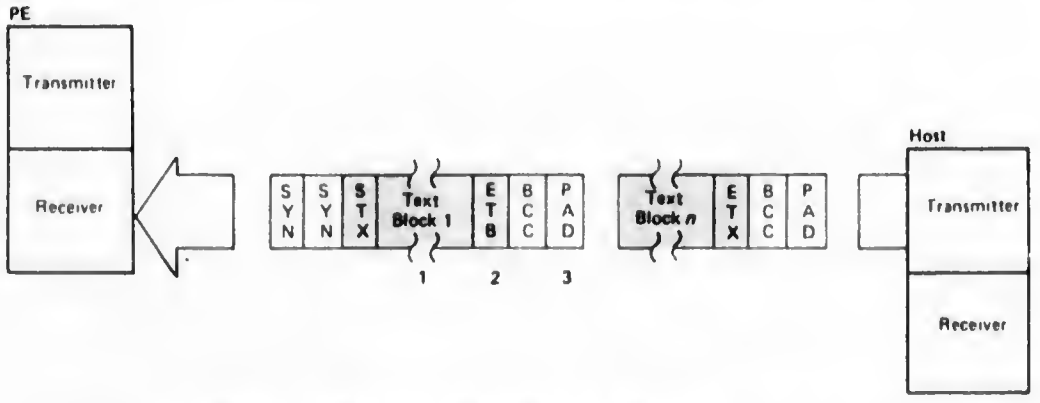
Data-Link Control Characters	Hexadecimal Bit Configurations	Function 1	Function 2	Function 3
EOT End of Transmission	37	Terminates the transmission of a message.	Resets the data link to Control Mode.	Indicates that a polled terminal has no data to transmit.
ENQ Inquiry	2D	Transmitter establishes initial contact with receiver.	Transmitter requests retransmit of response from receiver.	If transmitted in text, requests the receiver to ignore the message block.
NAK Negative Acknowledgement	3D	Receiver indicates that the last message block was incorrectly received.	Receiver requests that last block be retransmitted.	
ACK0(DLE 70) Positive Acknowledgement	1070	Receiver indicates that an even-numbered message block was received.	Receiver requests immediate transmission of next message block.	During selection, indicates that receiver is ready to receive data.
ACK1(DLE 61) Positive Acknowledgement	1061	Receiver indicates that an odd-numbered message block was received.	Receiver requests immediate transmission of next message block.	
WACK Wait before Transmit Positive Acknowledgement	106B	Receiver indicates that a message block has been successfully received	Receiver requests that transmission of next message block be delayed.	During selection, indicates that receiver is not immediately available, but will be in a short time.

Table 7-3(Cont): Functions and Hexadecimal Representations of BISYNC Message Interchange Control Characters

Data-Link Control Characters	Hexadecimal Bit Configurations	Function 1	Function 2	Function 3
TTD Temporary Text Delay	022D	Transmitter indicates that a positive acknowledgement of the previous block has been received.	Transmitter indicates a temporary delay before next block will be transmitted.	Forward abort-transmitter aborts current transmission.
RVI Reverse Interrupt	107C	Receiver tells transmitter that it has a high priority message to send.		

7.2.4 BISYNC Messages

In binary synchronous transmission, data, data-link control, and other characters are transmitted together in a contiguous series of bits. Figure 7-1 shows the structure of a typical BISYNC message and how the 3271 PE handles it.



Note. Shaded area indicate data that is recorded by the TRA program

1. Upon receiving data, the 3271 PE strips off protocol characters, takes the text block and directs text data to the 3270 TE or to the RSX user application program associated with the selected device address. The way the data is handled depends on the application program (see the RSX-11/3271 Protocol Emulator Programmer's Guide).

2. Long messages can be divided into a series of blocks. Each block of text, except for the last, should be followed by an ETB (an ETX follows the last block of text). The ETB requires the receiver to acknowledge successful or unsuccessful receipt of the message before the next block is transmitted.

3. The PAD characters at the end of the block are used to ensure that the last character is not truncated.

Figure 7-1: A BISYNC Message Stream

7.3 Message Formats and Operation Sequences

This section outlines message formats for the following 3271 operations initiated by the host system:

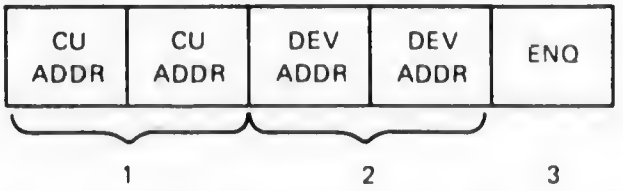
- Polling
- READ MODIFIED
- STATUS AND SENSE READ
- Selection
- WRITE

In addition to formats, this section illustrates the sequences observed for each operation by means of flow maps that chart interaction between the 3271 PE and the host system.

7.3.1 Polling

A polling sequence is sent by the host as an invitation to the polled PE station to send a message.

Message Format. The format of a polling sequence is as follows (numbers in the figure refer to the explanatory text):



1. Each 3271 PE control unit has a 1-character polling address (Table 7-4). To prevent errors, the address is given twice; if the addresses are not identical, the sequence is ignored.

2. Each station (logical display unit or printer) has a 1-character device address for specific polling (Table 7-4). All devices on a control unit can usually be polled under address 7F (general polling). To prevent errors, the address is given twice; if the addresses are not identical, the sequence is ignored.

3. The enquiry (ENQ) character indicates that this is a polling or selection sequence.

Table 7-4: Legal Polling Addresses for the 3271 Protocol Emulator

Control Unit or Device Number	EBCDIC Code (hex)	Control Unit or Device Number	EBCDIC Code (hex)	Control Unit or Device Number	EBCDIC Code (hex)
0	40	11	4B	22	D6
1	C1	12	4C	23	D7
2	C2	13	4D	24	D8
3	C3	14	4E	25	D9
4	C4	15	4F	26	5A
5	C5	16	50	27	5B
6	C6	17	D1	28	5C
7	C7	18	D2	29	5D
8	C8	19	D3	30	5E
9	C9	20	D4	31	5F
10	4A	21	D5	General Poll	7F

Operation Sequence for Poll. Figure 7-2 illustrates the operation of a polling sequence. Numbers in the figure refer to explanatory text that follows the figure.

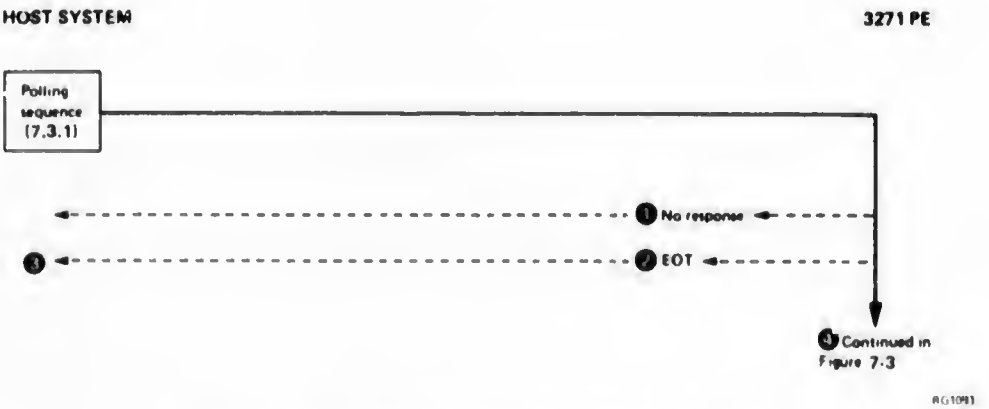


Figure 7-2: Operation Sequence for Poll

Notes:

❶ The 3271 PE fails to respond to a polling sequence and causes a timeout on the host system for any of the following reasons:

a. The 3271 PE is not loaded and running.

b. The polling sequence contains an invalid character.

c. The polling sequence is incomplete.

d. An address is incorrect.

❷ If there is no I/O pending or no device is in pending status, the 3271 PE issues an EOT. (Most devices are polled on a general poll.)

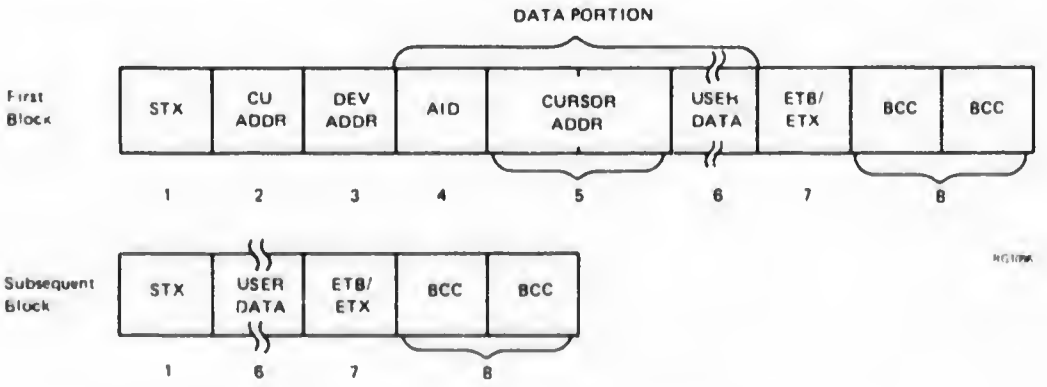
❸ The host system continues through its polling list for other devices.

❹ READ responses to polling sequences are shown in Figure 7-3.

7.3.2 READ MODIFIED Operation

A READ MODIFIED operation is initiated by a polling sequence from the host system. A READ MODIFIED operation takes the data sent to the 3271 PE by the RSX user application program (UAP), envelops it with the appropriate protocol, and sends it to the host system.

Message Format. The formats for the first and subsequent blocks of a READ MODIFIED message are as follows (numbers in the figure refer to explanatory text):



1. STX framing character provided by the 3271 PE.

2. Control unit address provided by the 3271 PE.

3. Device address provided by the 3271 PE.

4. Attention Identifier (AID) provided by the UAP as the first character in the data portion of the first transmit command. The AID character determines the type of data, or the type of action to take, on the incoming message.

5. Cursor address provided by the UAP as the second and third characters in the data portion of the first transmit command.

6. User data provided by the UAP as specified in the transmit command.

7. ETB or ETX framing character provided by the 3271 PE as specified by the end-of-data (EOD) indicator in the transmit command.

8. BCCs provided by the 3271 PE.

Items 4 and 5 are optional. They are included only if the user wants to emulate the IBM 3270 System. For more details, see Chapter 3 of the *RSX-11/3271 Protocol Emulator Programmer's Guide.*

A 3271 PE station (an I/O device simulated by an RSX user application program) may or may not be formatted, depending on the user data. Formatted data includes Set Buffer Address (SBA) orders. These orders define specific areas of the display screen (or printout) as data fields. In formatted displays, the location of the data fields is important. In unformatted displays, the location of the fields is not important. For more information on formatting user data to emulate a 3270 device, see Chapter 3 of the *RSX-11/3271 Protocol Emulator Programmer's Guide.*

Operation Sequence for a READ MODIFIED. Figure 7-3 illustrates the flow of operation resulting from a READ message response to a polling sequence. Numbers in the figure are keyed to explanatory text that follows the figure. This sequence is observed in the READ MODIFIED and the STATUS AND SENSE READ operations.

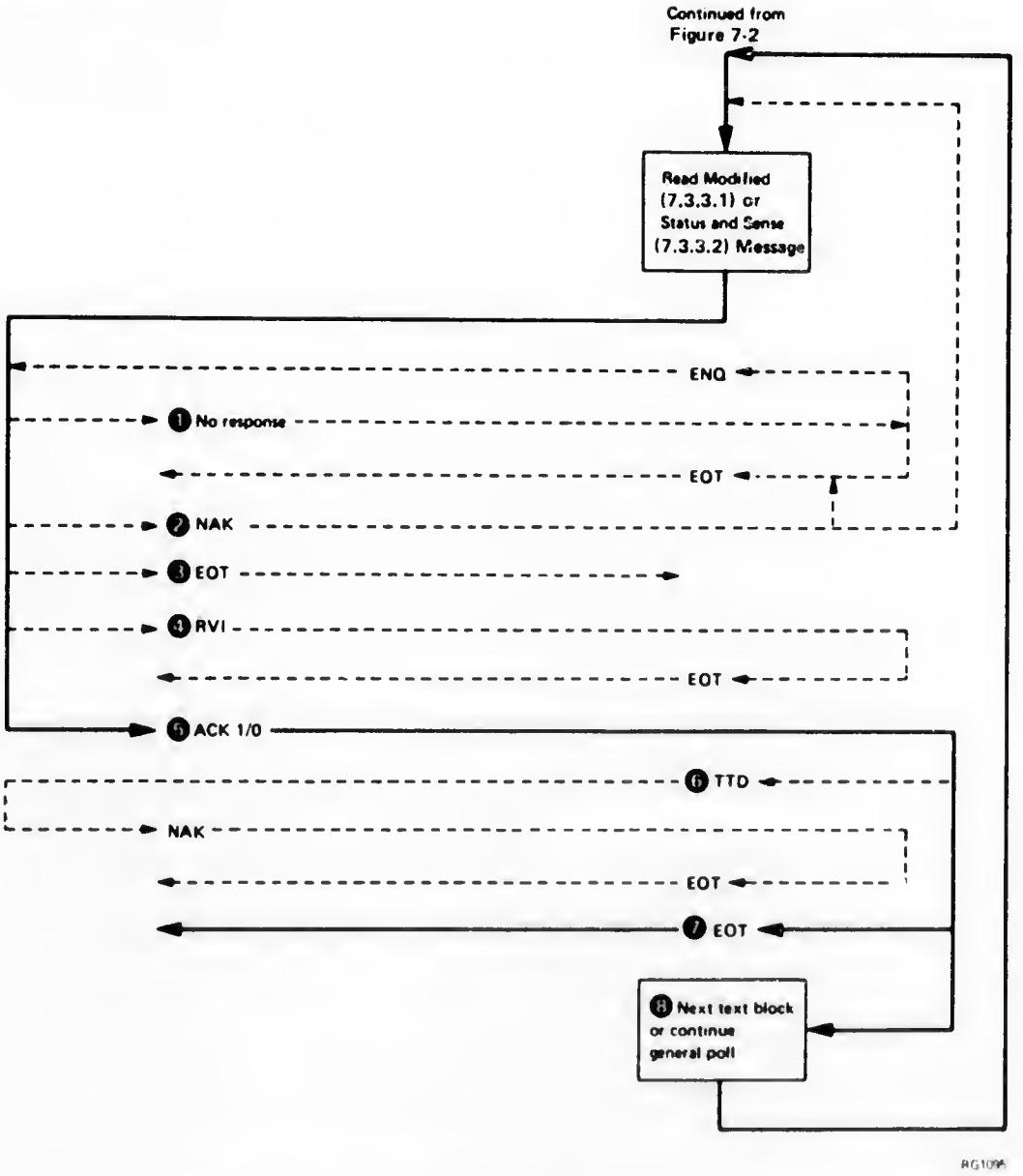


Figure 7-3: Operation Sequence for READ

Notes:

❶ A missing STX character caused transmission not to be received. After a 3-second timeout, the 3271 PE sends an ENQ to solicit a response from the host system. If the host does not respond after a specified number of ENQs (specified at PE generation), the 3271 PE aborts transmission with an EOT and informs the UAP of the termination. The number of ENQs permitted is specified in the 3271 PE Parameters subsection of the Configuration Database section of PEGEN.

❷ In this case, the host system has detected a BCC error or a missing ETB or ETX. The host's NAK response causes the 3271 PE to retransmit the last block of text. If the host responds with NAK a specified number of times (specified at PE generation), the 3271 PE aborts transmission with an EOT and informs the UAP of the termination. The number of NAKs permitted is specified in the 3271 PE Parameters subsection of the Configuration Database section of PEGEN.

❸ In this case, the host system aborts transmission if one of its error thresholds has been exceeded. The 3271 PE informs the UAP of the premature termination.

❹ In this case, the host system terminates the polling sequence with an RVI. The 3271 PE completes the termination with an EOT.

❺ Here, the host system acknowledges that the text block has been successfully received.

❻ In this case, the 3271 PE aborts transmission at the request of the UAP due to a system or hardware error.

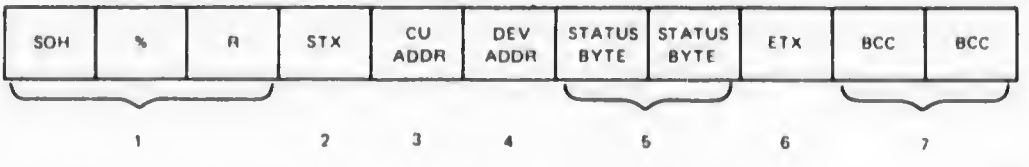
❼ Here, the 3271 PE terminates a general or specific poll with an EOT (normal termination).

❽ This shows a general poll (control unit address 7F), where the 3271 PE continues by sending data from the next available logical device until each device (with message or status pending) has sent a complete message (indicated by an ETX character). If there are several devices with data to send, the PE sends one message from each device, continuing through the polling list until all of these devices have sent their pending message.

7.3.3 STATUS AND SENSE READ

A STATUS AND SENSE READ returns to the host any status and sense information, instead of data, that has been maintained by the 3271 PE for the specified device(s). STATUS AND SENSE READ messages are identified by a special message header.

Message Format. The format for STATUS and SENSE messages is (numbers in the figure refer to explanatory text):



1. Status message header, provided by the 3271 PE

2. STX framing character, provided by the 3271 PE

3. Control unit address, provided by the 3271 PE

4. Device address, provided by the 3271 PE

5. Status and sense bytes (2), provided by the 3271 PE (Table 7-5)

6. ETX framing character, provided by the 3271 PE

7. BCCs provided by the 3271 PE

The significance of the possible status settings is given in Table 7-5. The *IBM 3270 Information Display System Component Description* manual fully describes sense/status conditions and corresponding bit patterns.

When an event listed in Table 7-5 occurs, the designated sense/status bits are set and the information is stored by the PE until the device is polled. When the device is polled (by general or specific poll), the PE sends any pending sense/status information for that device to the host. How the host reacts to the information received in the status/sense bits depends on the host application.

Table 7-5: Status and Sense Byte Definitions for 3271 Protocol Emulator

Byte Number	Bit Number	Name	Meaning
0	7	—	Set if bit 1 is set.
	6	—	Always set to one.
	5	—	Reserved.
	4	—	Reserved.
	3	—	Reserved.
	2	—	Reserved.
	1	DE (Device End)	When set, indicates that the status of the addressed device has changed from off-line to on-line, or from busy to not busy.
	0	—	Reserved.
1	7	—	Never set.
	6	—	Always set to one.

Table 7-5 (Cont): Status and Sense Byte Definitions for 3271 Protocol Emulator

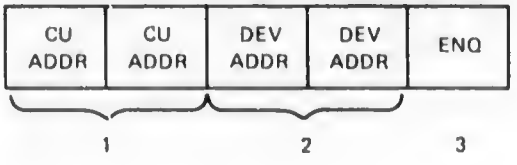
Byte Number	Bit Number	Name	Meaning
1	5	CR (Command Reject)	When set, indicates that the 3271 PE has received a COPY or READ BUFFER command from the host. These commands are not supported by the 3271 PE. The READ BUFFER command may be supported if your application program has attached the associated TU device with the AST option.
	4	IR (Intervention Required)	When set, indicates that manual intervention is required to resolve a situation where a selection sequence or specific poll was directed to a device that is unavailable or not online.
	3	—	Reserved.
	2	—	Reserved.
	1	—	Reserved.
	0	—	Reserved.

Operation Sequence. The operation sequence is identical to that of the READ MODIFIED operation, illustrated in Figure 7-3.

7.3.4 Selection

The host sends a selection sequence to a PE station to demand that the station receive a message from the host.

Message Format. The format for a selection sequence is as follows (numbers in the figure refer to explanatory text):



1. Each 3271 PE control unit has a 1-character selection address (Table 7-6). To prevent errors, the address is given twice; if the addresses are not identical, the sequence is ignored. Except control unit numbers 1 and 16, the selection address is determined by adding 20 (hexadecimal) to the unit's polling address.

2. Each logical 3271 PE device has a 1-character device address (Table 7-4) for selection. To prevent errors, the address is given twice; if the addresses are not identical, the sequence is ignored.

3. The ENQ character indicates that this is a polling or selection sequence.

Table 7-6: Legal Selection Addresses for the 3271 Protocol Emulator

Control Unit Number	Graphic Address Character	EBCDIC Code (hex)	Control Unit Number	Graphic Address Character	EBCDIC Code (hex)
0	-	60	16	0	F0
1	/	61	17	1	F1
2	S	E2	18	2	F2
3	T	E3	19	3	F3
4	U	E4	20	4	F4
5	V	E5	21	5	F5
6	W	E6	22	6	F6
7	X	E7	23	7	F7
8	Y	E8	24	8	F8
9	Z	E9	25	9	F9
10	\	6A	26	:	7A
11	,	6B	27	#	7B
12	%	6C	28	@	7C
13	UNDERSCORE	6D	29	'	7D
14	)	6E	30	=	7E
15	?	6F	31	"	7F

Operation Sequence for Selection. Figure 7-4 illustrates the flow of operation in response to a selection. Numbers in the figure are keyed to explanatory text that follows the figure.

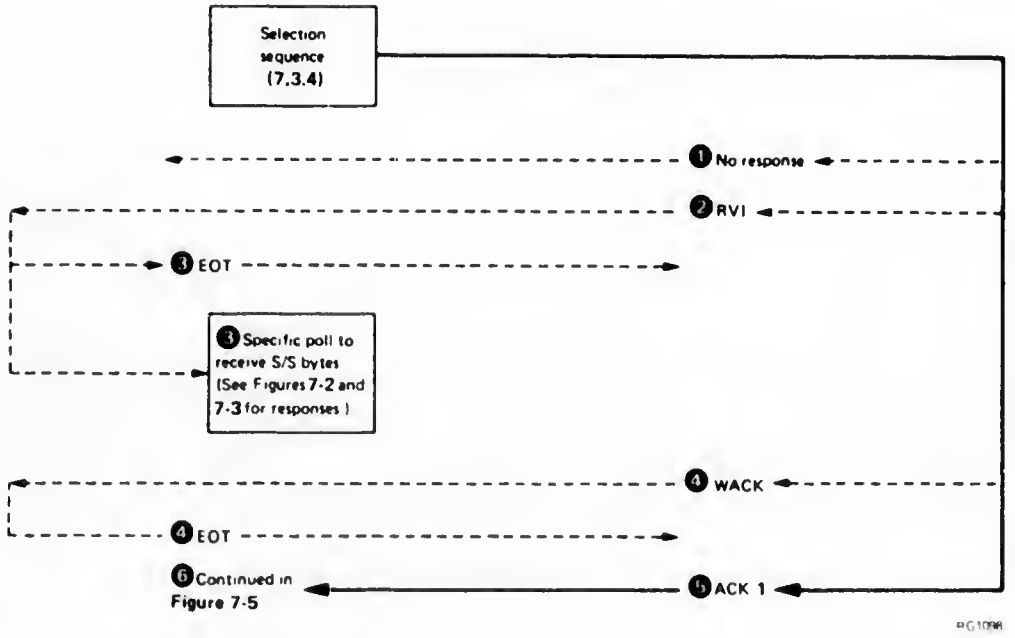

Selection sequence (7.3.4)
1 No response
2 RVI
3 EOT
3 Specific poll to receive S/S bytes (See Figures 7-2 and 7-3 for responses.)
4 WACK
4 EOT
6 Continued in Figure 7-5
5 ACK 1

Figure 7-4: Operation Sequence for Selection

Notes:

❶ The 3271 PE fails to respond to a selection sequence and causes a timeout on the host system for any of the following reasons.

a. The 3271 PE is not loaded and running.

b. The selection sequence contains an invalid character.

c. The selection sequence is incomplete.

d. An address is incorrect.

e. The communications line or modem is not functioning properly.

❷ In this case, the addressed device is unavailable or has a status message (other than Device End) to be transmitted.

❸ In this case, the host system terminates the selection sequence with an EOT and issues a specific poll to the device that has the pending status. The possible responses to this poll are shown in Figures 7-2 and 7-3.

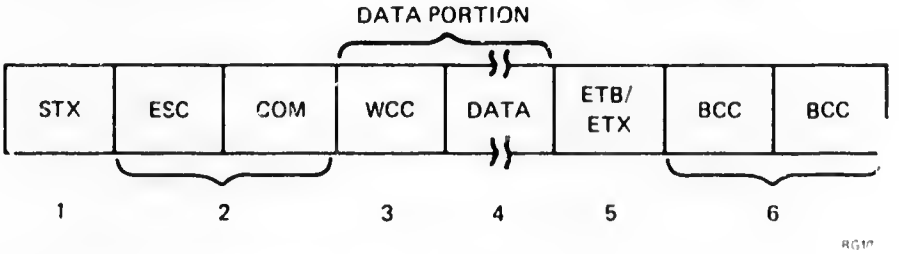
❹ In this case, the 3271 PE acknowledges that the selection address has been successfully received and that there is no status pending for the device.

❺ The WRITE command sequence and responses are shown in Figure 7-5.

7.3.5 WRITE Operation

The host system initiates the WRITE operation. On detecting a WRITE operation from the host, the 3271 PE takes the incoming message, strips the appropriate protocol from the incoming message, and directs the data to the selected PE station (RSX user application program).

Message Format. The format for the WRITE command message is as follows (numbers in the figure refer to explanatory text):



1. STX framing character.

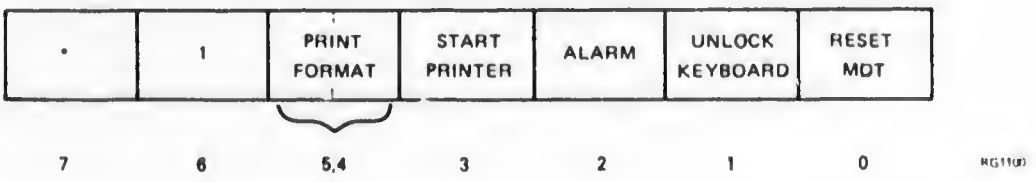
2. Escape sequence specifying the type of WRITE command initiated by the host system. The command type (COM) and any associated data are passed to the UAP to await any further action. The only exception to this is ESC 6, which causes the 3271 PE to perform a normal READ MODIFIED operation.

   The UAP can assign any significance to each ESC command. The standard 3270 commands are:

   ESC 1 WRITE
   ESC 5 ERASE/WRITE
   ESC ? ERASE ALL UNPROTECTED
   ESC 6 READ MODIFIED

   These commands and 3270 emulation are discussed in detail in the *RSX-11/3271 Protocol Emulator Programmer's Guide*.

3. The Write control character is optional, and is used for the 3270 device emulation only. The format is as follows:



Bit 7 (ignore)	Determined by contents of bits 5 to 0
6	Always set (ignore)
5,4	Indicates printout format:

00 NL order determines line length
01 40-character line
10 64-character line
11 80-character line

3	If set, start printer.
2	If set, sound audible alarm
1	If set, restore keyboard enable.
0	If set, reset MDT bits.

When used, the WCC is the first character in the user receive data buffer.

4. Data portion of message that is passed to the UAP.

5. ETB or ETX framing character.

6. BCCs.

A 3271 PE station (an I/O device simulated by an RSX user application program) may have either a formatted display or an unformatted display, depending on the user data. User data for a formatted display includes Start Field (SF) orders. These orders define specific areas of the display screen (or printout) as data fields, having certain properties (or attributes). For detailed information on formatting data, see Chapter 3 in the *RSX-11/3271 Protocol Emulator Programmer's Guide.*

Operation Sequence for WRITE. Figure 7-5 illustrates the sequence observed when a WRITE command is issued in response to a selection sequence. Numbers in the figure are keyed to explanatory text that follows the figure.

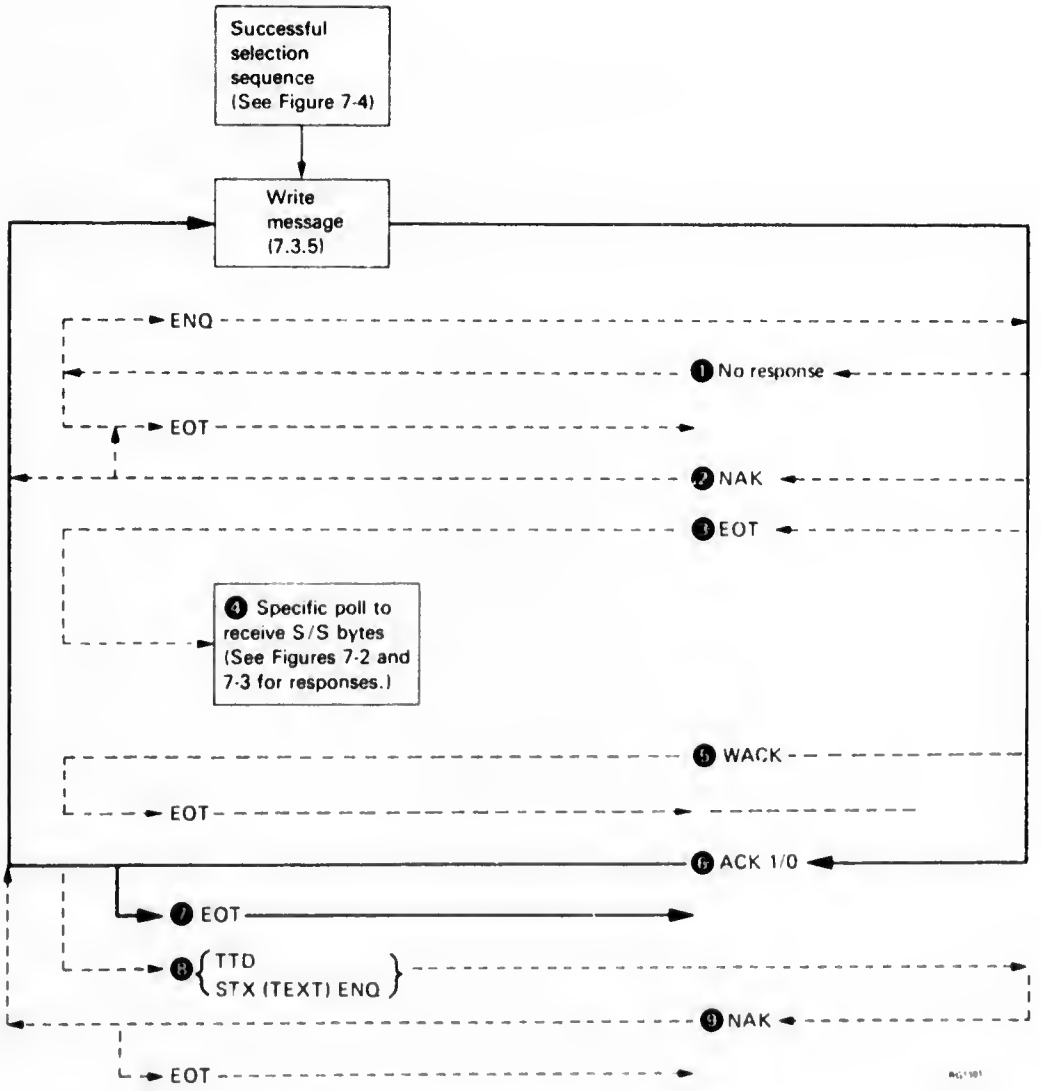


Figure 7-5: Operation Sequence for WRITE

Notes:

❶ Transmission was not received because of a missing STX character. After a 3-second timeout, the host sends an ENQ to solicit a response from the 3271 PE. If the 3271 PE does not respond after a specified number of ENQs (as specified at PE generation), the host aborts transmission with an EOT.

❷ The 3271 PE has detected a BCC error or a missing ETB or ETX. The 3271 PE NAK response causes the host system to transmit the last block of text. If the 3271 PE responds with a NAK a certain number of host-specified times, the host aborts transmission with an EOT.

❸ The 3271 PE cannot perform the operation specified in the command because the logical unit is closed or because the command is invalid.

❹ The host system issues a specific poll (to the device that rejected the selection sequence) to obtain the status/sense bytes. The possible responses to this poll are shown in Figures 7-2 and 7-3.

❺ The 3271 PE acknowledges that the text block has a valid command and the BCC is valid.

❻ The host system issues an EOT to terminate the selection sequence (normal termination).

❼ The host system sends a temporary text delay or performs a forward abort.

❽ The 3271 PE sends a NAK in response to a TTD or a text block containing an ENQ. If it was a TTD, the host retransmits the last block. If it was a forward abort, the host terminates the selection sequence with an EOT and the 3271 PE informs the UAP.

Chapter 8
Error Handling and Troubleshooting

In Chapter 5 you were shown how to test your PE system to make sure that it operates properly. This chapter describes the 3271 PE error handling capabilities, and tools and procedures that will help you isolate and identify communications problems that may arise during use of the PE. This chapter also lists the error messages that may be generated by the various PE components.

8.1 Error Handling

There are three types of errors, discussed in the following subsections, that may occur when using the 3271 PE:

1. **Installation and reloading errors.** These may arise when loading or manipulating the PE.

2. **Runtime communications errors.** These may arise after the PE has been installed and verified.

3. **Component errors.** These may arise when using those components that perform functions other than loading or manipulating the system.

8.1.1 Installation and Reloading Errors

Errors that occur when loading or manipulating the system are generated by utilities such as the Network Loader (NTL) and Network Initializer (NTINIT). NTL and NTINIT load and unload software in response to NCP commands. Error messages are displayed on the requesting terminal, and remedial actions to be taken are generally self-evident. These errors usually indicate compatibility mismatches between certain RSX system components and PE components. Some examples of this error type indicate vectors that are unavailable or system-controlled partitions that are too fragmented.

8.1.2 Runtime Communications Errors

Errors that occur during operation of the 3271 PE arise from problems with either the communications equipment or with the user application program. Equipment that may fail includes modems, controller devices, and communications lines. Software problems may arise from mistakes such as incorrect addressing of control units or insufficient buffer allocation. (Section 8.3 provides suggestions for troubleshooting hardware and software errors.) The 3271 PE automatically handles most software and hardware problems.

One of the 3271 PE's principal responsibilities is to ensure the integrity of data transmitted over the communications link. To detect and correct BISYNC transmission errors, the 3271 PE uses the cyclic redundancy check method of error handling (see Section 7.1). When the PE detects a faulty message block, it requests the host to retransmit the block. If, after retransmission, the message block is still erroneous, another transmission is requested. This cycle continues until the message block is received without error or until the retransmission count exceeds the NAK threshold count specified during PEGEN. In the latter case, an EOT response is sent to the host, causing the host to abort further transmission of that message. The host must then make the appropriate error recovery response. The PE records all errors on a line, but does not bring a line down.

When a faulty message sent by a PE is replied to with a NAK too many times by the host, the host may shut down the station or line involved and report the error to the operator. The host's response depends on the system used by the host. In any case, the RSX user receives an error message and is responsible for retransmitting the message correctly or taking proper error-recovery actions.

Other error-handling features in the 3271 PE include:

- Attempting recovery from potentially fatal errors by informing the user programs through return codes and system messages. The 3271 PE is built to prevent fatal errors. However, conditions can arise that cause messages to be lost or transmissons to be aborted.

- Maintaining statistics on the number of recoverable error counts (such as the number of NAKs transmitted) and the number of error-count thresholds that are exceeded (such as the number of NAK thresholds exceeded). These statistics are useful for evaluating line and physical link performance and they help alert you to possible line problems.

- A Trace (TRA) option permitting you to examine the actual protocol messages on a specific line.

- A loopback test program (LBK) permitting you to test hardware configuration and to diagnose problems.

Error-handling information provided by the emulator can give you the information you might need to incorporate error-handling procedures in your application programs. For example, if the 3271 PE returns a code indicating that I/O buffers are full and that it cannot honour a pending request to transmit another message, the user program should wait until a prior request is completed and its buffer is released. See the *RSX-11/3271 Protocol Emulator Programmer's Guide* for further details on (QIO) error codes and error-handling procedures.

(Section 8.2 discusses the utilities that provide statisics.)

8.1.3 Component Errors

Errors that occur during use of the various PE utilities are generated by the utility being used. However, when using NCP for loading or unloading software, the error may be generated by a subsidiary component, as explained above. Component errors are listed and explained in Section 8.4.

8.2 PE Troubleshooting Tools

The 3271 PE provides tools, described in the following subsections, which are used to isolate and identify communications problems:

- NCP line monitoring function
- Error Logging and Statistics Routine (DCE)
- Trace program (TRA)
- Line Loopback Test (LBK)
- Network Dump Analyzer (NDA)

8.2.1 Using NCP

NCP's SHOW LINE COUNTERS command gives you access to statistics, which you can use to localize the cause of communications failures. These statistics are provided by a set of counters maintained by the PE's BSD module. Section 6.4.2.1 explains how to use the SHOW LINE COUNTERS command and gives display examples. The significance of each counter is explained in Section 6.4.2.2, in the discussion of the DCE utility. The DCE utility provides statistics on the same counters as does NCP.

Problems on a line may be indicated when certain counters show unusually high counts. For example, a large number of NAKs transmitted may indicate that the host facility is having problems with its communications hardware or lines. Perhaps lines are experiencing noise interference, or modems are malfunctioning.

8.2.2 Using the Error Logging and Statistics Routine (DCE)

DCE provides formatted reports of the statisics maintained by the BSD module. These are the same statistics as displayed by NCP's SHOW LINE COUNTERS command. These reports may be kept in a log file. Section 6.4.2.2 explains how to use DCE and shows report examples. DCE's troubleshooting usage is similar to that of NCP, described above.

8.2.3 Using The Trace Program (TRA)

The Trace program (TRA) is an optional facility that you may select during PEGEN. It is highly recommended for troubleshooting and for testing and debugging the communications aspect of user applications. This option is required to run loopback tests: see section 8.2.4. If you select this option, the line trace feature is built into the PE. The PE assigns a Trace unit to each line to retrieve all traffic, including user data, communications and protocol information. (The lowest TU address for each line is reserved for a TRA unit. See Section 3.5.)

The 3271 PE treats each TRA unit as a pseudodevice on a controller unit. Each TRA unit is assigned trace buffers by the TRA program. (In addition, you can have application programs assigning trace buffers to each unit. These programs can be used in place of the TRA program, but not at the same time.) Trace buffers store all data transmitted and received on the line, providing that the user has supplied a sufficient number of buffers. The TRA program generates formatted reports of the data as it appeared on the line to the PE. The TRA output contains untranslated EBCDIC data and all necessary communication software protocol messages (see Figure 7-1 and Figures 8-1 and 8-2).

8.2.3.1 Trace Program Operation — The 3271 PE places messages into a trace buffer. When the buffer is full, the PE returns a QIO completion to the TRA program. If there is more traffic, the PE uses another trace buffer. If you are using your own application program for gathering and reporting data received by TRA, make sure that it provides a sufficient number of buffers. If another buffer is not available, the PE cannot save more data. To ensure efficient trace processing and to prevent loss of trace data, the application program should, therefore, assign at least three buffers, each capable of holding four complete messages.

To collect and examine the trace data, a UAP must post an ATTACH QIO and READ QIO for the appropriate Trace unit (QIO Macro user interfaces are discussed in the *RSX-11M/3271 Protocol Emulator Programmer's Guide*).

To distinguish the received data from the transmitted data, TRA prefixes each message by a unique 2-byte octal code. The code is transparent to the user, but is translated in the TRA program report. For received messages, the code is

First byte: 020

Second byte: 331

For transmitted messages, the code is

First byte: 020

Second byte: 343

The EBCDIC data follows the 2-byte code.

The Trace program may be run either from MCR, or installed and invoked, as shown in Section 8.2.3.3.

8.2.3.2 Trace Program Commands — The Trace program uses a standard RSX command line to specify the output file and switches for the report process, and the TU input device. The command line observes the following format:

dev:[*g,m*]*filename.type/switch1/switchn*=Tu*n*:

where

dev:	is the physical or logical device unit on which the volume containing the output file is mounted. The device unit is expressed as a 2-character ASCII device name and an optional 1- or 2-digit octal unit number, followed by a colon (DK0:, DT1: or LP:, for example).
[*g,m*]	is the UIC that specifies the User File Directory in which the file is listed. The variables *g* and *m* are octal numbers from 0 to 377 that represent the owner's group and member numbers, respectively. Brackets are a mandatory part of the UIC.
filename...	is the name of the file, an alphanumeric string from 1 to 9 characters in length. The default file name is TUPRT.
.type	is a 3-letter mnemonic preceded by a dot. It identifies the nature of the file's contents. The default is TXT for this program.

/switch consists of a slash and 2 characters that identify a switch option. You can modify the switch name with decimal or octal numbers. Specify these numbers by using the following conventions:

- Decimal numbers are followed by a decimal point (for example 3.).
- Octal numbers are preceded by a pound sign (for example, #3).

  When octal or decimal is not specified, the default is decimal.

Valid switches are:

/SH specifies the short output line of 72 (decimal) characters as opposed to the normal print line of 132 (decimal) characters. This switch is used for tracing messages to a video terminal or a short line printer. It can also save space on a disk file because the lines it creates are shorter. See Figure 8-1.

/PA:*n* specifies the number of output pages after which the trace is to terminate. This number is approximate only, and the trace will terminate the next time a trace buffer is presented after *n* pages. The number of pages is *n*; the default is 200 (decimal) pages.

/TI:*n* specifies the time, in minutes, the trace is to operate. At the end of this time period, the trace terminates. The default period is 5 minutes.

= is the delimiter that separates the input and output file specifications in the command string.

TU*n*: specifies the number of the 3271 PE Terminal Unit (for example: TU0 or TU23) to be used for the trace function. The two ASCII characters, TU, are required; *n* may be any valid 2-digit octal number. The octal number is the lowest address of all the stations on the emulated control unit.

8.2.3.3 Trace Program Example — The Trace program (TRA) may be run or installed from MCR, as shown below.

Example 1. Using MCR

```
)RUN TRA
TRA)DK:FILE.TXT/PA:50/SH/TI:10=TU0:
```

With this command line, the Trace program directs the output to a disk file named FILE.TXT on device DK:. The program will terminate after 50 pages are output (/PA:50); the length of each output line will be 72 characters (/SH). The trace function will run for a period of 10 minutes (/TI:10), unless the 50 pages of output are produced in a shorter time period. When both the /PA and /TI switches are used, TRA will terminate according to whichever condition occurs first: the reaching of designated number of ouput pages, or the expiration of the designated time period. The Trace unit for input (=TU0:) is the one assigned to the control unit with the lowest address as specified during PEGEN.

Example 2. Installing and invoking TRA

```
)INS |167,54|TRA
)TRA

TRA)TI:/SH/TI:3=TU7:
```

This command line instructs the Trace program to list the output on the terminal (TI:) in short output lines (/SH) for a time period of three minutes (/TI:3). The Trace unit is TU7, the seventh TU device known to the PE.

8.2.3.4 Interpreting Retrieved Messages — Examples of output from the Trace program are given in Figures 8-1 and 8-2. The typical trace output line has three parts:

1. RECV, XMIT, or blank, RECV and XMIT indicate the direction of the data flow with respect to the PE. A blank indicates that the line is a continuation of the previous line.

2. The data, represented in ASCII. A period (.) replaces any nonprinting value, such as communication control characters.

3. The data, represented in hexadecimal.

A trace output line that begins with BUFFER=B*n* indicates that the *n*th buffer has been completely formatted.

In Figure 8-1 several lines of output are annotated. The numbers, located in the lefthand column, refer to the explanatory notes that follow:

❶ The PE is receiving a general poll. The address, as known to the host, of the emulated controller is hexadecimal 40, which corresponds to STA0 (TU0:). The general poll is indicated by hexadecimal 7F. The 2D is an inquiry character (ENQ) indicating that this message is a poll. See Section 7.3.1 for a detailed discussion of the polling message format.

❷ The PE reponds by transmitting a message. The first character is the Start of Text character (STX), hexadecimal 02. This is followed by the controller address and the data. The end of the message is indicated by the End of Text (ETX) character, 03.

❸ The PE receives a positive acknowledgement from the host, hexadecimal 61.

❹ The PE transmits an End of Transmission (EOT) message, hexadecimal 37. This signifies to the host that the PE has no further data to transmit.

❺ The PE receives an End of Transmission message from the host. The PE returns to control mode, waiting to be polled again by the host.

The Trace program uses three 512-byte buffers for gathering the raw data from the 3271 PE Trace unit. If the Trace program does not output data at a rate faster than the rate at which the data is being gathered, it is possible to lose data during the trace operation. The loss of data is evident within one line of the message signifying that the buffer has been completely formatted.

Use the Translation Tables provided in the *RSX-11/3271 Protocol Emulator Programmer's Guide* to translate the EBCDIC characters into ASCII.

```
3271PE TRACE EXAM2/SH/PA:1=TU:                16-OCT-80 16:19:18 PG     1
RECV   /                          61
XMIT   .                          37
RECV   .                          37
RECV     "".                      40407F7F2D
XMIT   .   B$$N2 = 000002  .      024040C25B5BD5F2407E40F0F0F0F0F0F240404005
       LD$DB = 000000    .R$$     D3C45BC4C2407E40F0F0F0F0F0F040404005D95B5B
       NDC= 000005     .SYNCNT  D5C4C37E40F0F0F0F0F0F540404005E2E8D5C3D5E3
       = 000003 G .U$$MLO= 1    7E40F0F0F0F0F0F340C74005E45B5BD4D3D67E40F1
       60000.                   F6F0F0F0F003
RECV   /                          61
XMIT   .                          37
RECV   .                          37
RECV     "".                      40407F7F2D
XMIT   .  B$$N3 = 000002   .      024040C25B5BD5F3407E40F0F0F0F0F0F240404005
       LD$DK = 000000    .R$$     D3C45BC4D2407E40F0F0F0F0F0F040404005D95B5B
       NDH= 000226    .S$$ECC   D5C4C87E40F0F0F0F2F2F64040405E25B5BC5C3C3
       = 000000    .U$$MRN= 1   7E40F0F0F0F0F0F040404005E45B5BD4D9D57E40F1
       70234.                   F7F0F2F3F403
RECV   /                          61
XMIT   .                          37
RECV   .                          37
❶RECV    "".                      40407F7F2D
❷XMIT  .  C$$CDA= 000002   .      024040C35B5BC3C4C17E40F0F0F0F0F0F240404005
       LD$DM = 000000    .R$$     D3C45BC4D4407E40F0F0F0F0F0F040404005D95B5B
       NDL= 000001    .S$$TOP   D5C4D37E40F0F0F0F0F0F140404005E25B5BE3DGD7
       = 000000    .V$$CTR= 0   7E40F0F0F0F0F0F040404005E55B5BC3E3D97E40F0
       01000.                   F0F1F0F0F003
❸RECV  /                          61
❹XMIT  .                          37
❺RECV  .                          37
RECV     "".                      40407F7F2D
XMIT   .  C$$CKP= 000000   .      024040C35B5BC3D2D77E40F0F0F0F0F0F040404005
       LD$DT = 000000    .R$$     D3C45BC4E3407E40F0F0F0F0F0F040404005D95B5B
       SND= 000000    .S$$WPC   E2D5C47E40F0F0F0F0F0F040404005E25B5BE6D7C3
       = 000036    .V$$RSN= 0   7E40F0F0F0F0F3FG4040405E55B5BD9E2D57E40F0
       00032.                   F0F0F0F3F203
RECV   /                          61
XMIT   .                          37
RECV   .                          37
RECV     "".                      40407F7F2D
XMIT   .  C$$CSR= 177404   .      024040C35B5BC3E2D97E40F1F7F7F4F0F440404005
       LD$LP = 000000    .R$$     D3C45BD3D7407E40F0F0F0F0F0F040404005D95B5B
BUFFER= B1 000001 001000
       11M= 000000    .S$$WP    F1F1D47E40F0F0F0F0F0F040404005E25B5BE6D7
```

Figure 8-1 A Trace Program Report, Short Form (72-character lines)

```
3271PE TRACE E'AMPLE·PA:5=TU:                    16·OCT 8· 16:17:4 PG   3

      LTIPLE'ER UNIT NUMBER.                      D3E3C9D7C3C5E7C5D94·E4D5C9E34·D5E4D4C7C5D903
RECV  /              61
 MIT  .              37
RECV  .              37
RECV    "".                                      404·7F7F2D
XMIT  .  .000064.   000 .....B·TE.0.: ·A  : NU 024·4·05F0F0F·F·FGF4·54·4·4·0F·F0F0·4·05·5·5·548C2E8E3C5·5F·0·55E4·4DC·5D4·05E4·0D5E4
      MBER OF TRIBUTARIES ON THIS LINE.        D4C2·5D94·D6C64·E3D9F9C2E4E2C1D9C5E34·D6D54·E3C8C9C34·D3F9F7C503
RECV  /                                        61
 MIT  .                                        37
RECV  .                                        37
RECV    "".                                    404·7F7F2D
XMIT  .  .000065.   001 .....B·TE.1.: ·D· : LI 024·4·05F0F0F·F·FGF5·54·4·4·0F·F·F14·0·5·5·5·548C2E8E3C5·5F1·55E4·4DC45D4·05E4·0D3C9
      NE COST FOR ROUTING TASK.                D5C54·C3D6E2E34·C6D6D94·D9D6E4E3C9D5C74·E7C1E2D203
RECV  /                                        61
XMIT  .                                        37
RECV  .                                        37
RECV    "".                                    404·7F7F2D
XMIT  .  .000066.000000 .....WORD.0.: (D) : PO 024·4·05F0F0F0F0FGF6·5F·F·F0F·F·F·4·0·5·5·5·548E6D6D9C4·5F·0·55E4·4DC45D4·05E4·0D7D6
      INTER TO CONTROLLER REQUEST BLOCK.       C3D5E3C5D94·E3D64·C3D6D5E3D9D6D3D3C5D94·D9C5D8F4C5E2E34·C2D3D6C3D203
RECV  /                                        61
XMIT  .                                        37
RECV  .                                        37
RECV    "".                                    404·7F7F2D
BUFFER= 83 000001 001000
XMIT  .  .000070.000000 .....WORD.0.: (D) : NE 024·4·05F0F0F0F0F7F0·5F·F·F0F0F·F·4·0·5·5·5·548E6D6D9C4·5F·0·55E4·4DC45D4·05E4·0D5C5
      TWORK MANAGEMENT LINE STATE.             E3E6D6D9D24·D4C1D5C1C7C5D4C5D5E34·D3C9D5C54·E2E3C1E3C503
RECV  /                                        61
XMIT  .                                        37
RECV  .                                        37
RECV    "".                                    404·7F7F2D
XMIT  .  ..000000 .....REPT.0.                 024·4·05·5F0F0F0F·F·0F·4·0·5·5·5·548D9C5D7E3·5F·03
RECV  /                                        61
XMIT  .                                        37
RECV  .                                        37
RECV    "".                                    404·7F7F2D
XMIT  .  .......BYTE.0.: (D) : MULTIPOINT TRIB 024·4·05·5·5·5·5·548C2E8E3C5·5F·0·55E4·4DC45D4·05E4·0D4E4D3E3C9D7C6C9D5E34·0E3D9C9C2
      UTARY FLAGS.                             E4E3C1D9E84·0C6D3C1C7E203
RECV  /                                        61
XMIT  .                                        37
RECV  .                                        37
RECV    "".                                    404·7F7F2D
XMIT  .  .......BYTE.1.: (D) : TRIBUTARY ROUTI 024·4·05·5·5·5·5·548C2E8E3C5·5F1·55E4·4DC45D4·05E4·0E3D9C9C2E4E3C1D9E84·0D9D6E4E3C9
      NG COST.                                 D5C74·0C3D6E2E303
RECV  /                                        61
XMIT  .                                        37
RECV  .                                        37
RECV    "".                                    404·7F7F2D
XMIT  .  .......WORD.0.: (D) : NETWORK MANAGEM 024·4·05·5·5·5·5·548E6D6D9C4·5F·0·55E4·4DC45D4·05E4·0D5C5E3E6D6D9D24·0D4C1D5C1C7C5D4
      ENT TRIBUTARY STATE.                     C5D5E340E3D9C9C2E4E3C1D9E84·0E2E3C1E3C503
RECV  /                                        61
XMIT  .                                        37
RECV  .                                        37
RECV    "".                                    404·7F7F2D
XMIT  .  .......ENDR.                          024·4·0·5·5·5·5·5·548C5D5C4D903
RECV  /                                        61
```

Figure 8-2 A Trace Program Report, Long Form (132-Character Lines)

8.2.4 Loopback Test Program

The Loopback Test program (LBK) enables you to test BU and DV lines, examine modem control signals, and protocol messages received from the host system.

NOTE

LBK is not supported for a KBU line. When you have problems with a KBU controlled line, you should stop the line, clear the KBU, load the BU driver, and retry the operation. Then you can do loopback testing on the line.

In order to use LBK, you must have selected the Trace option during PEGEN; in order to run it, you must first stop all user application program traffic on the lines to be tested (lines must remain turned ON), and then manually loop back the line hardware. You can loop a line back at the device interface, in the modem, or on the EIA cable with a cable loopback device.

NOTE

Be careful not to exceed EIA specifications for line length. A passive loopback connector, like the H325, requires that the signals from the line interface travel twice the distance of the cable.

A line may also be looped back in the modem, if the modem supports loopback functions. Figure 8-3 shows the hardware that can be tested by LBK. Figures 8-4 through 8-6 show how the hardware is arranged to perform the loopback tests.

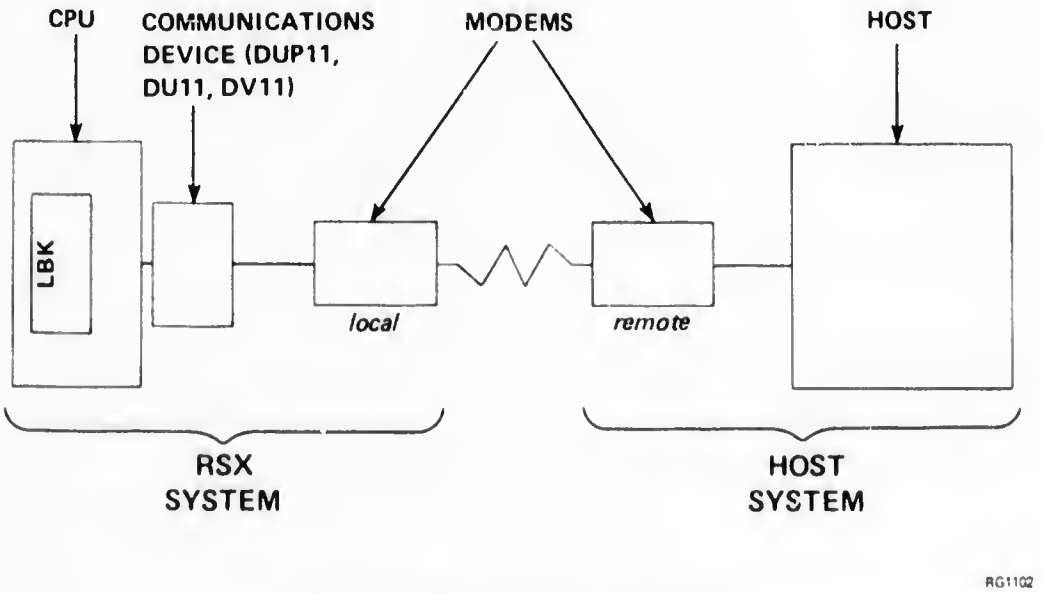


Figure 8-3: Systems under Test

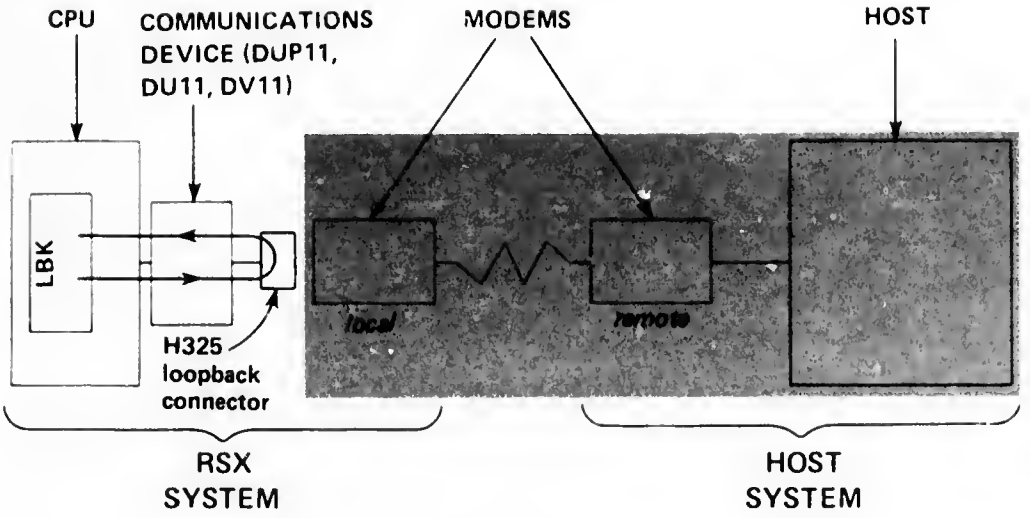


Figure 8-4: Line Loop Test

Line loop test (Figure 8-4). The loopback connector is installed on the cable between the communications device and the modem. This arrangement tests the communications device and the cable. The connector returns digital data to the communications device. The line loop test can be performed on DUP11 and DV11 line interfaces only.

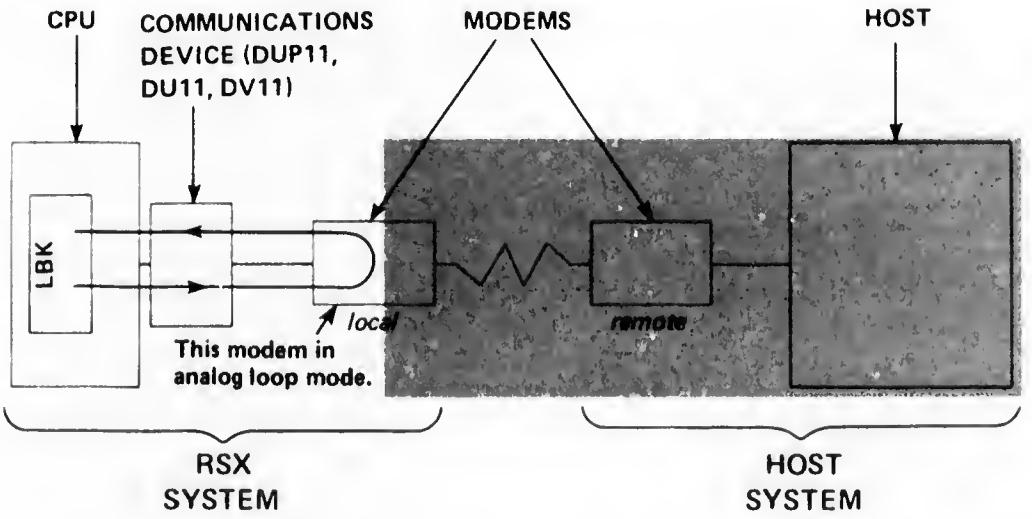


Figure 8-5: Modem Loop Test (Local)

Local modem loopback test (Figure 8-5). This test can be performed if the local modem has a data-loopback test feature. For example, Bell 208 modems have an Analog Loopback (AL) button that directs the modem's transmitter output to its receiver input. The loopback arrangement tests the communications device, the cable, and the modem. The modem converts data from the communications device to analog form, digitizes the data again, and returns it to the device. The line must be in a full-duplex mode.

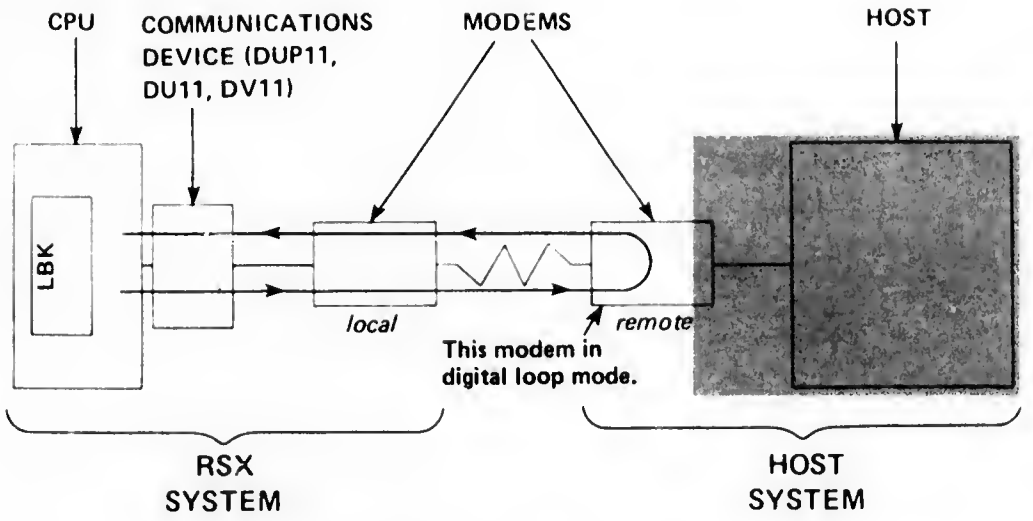


Figure 8-6: Modem Loop Test (Remote)

Remote modem loopback test (Figure 8-6). This test can be performed if the remote modem has a digital loopback feature (transmits back any data received on the line). For example, Bell 208A modems have a Digital Loopback (DL) button that directs the modem's receiver output to its transmitter input. This loopback arrangement tests the communications device, the cable to the local modem, the local modem, the telephone circuit, and the remote modem. The remote modem digitizes data from the local modem, converts the data to analog form again, and returns it to the local modem and finally to the communications device. The line must be in full-duplex mode.

To run LBK, enter;

)LBK

The prompt, LBK) will then appear, and you are ready to enter any of the commands described in Section 8.2.4.1. Listings and messages will appear at the calling terminal.

You can also write your own program to perform loopback testing. Section 8.2.4.3 describes program implementation.

8.2.4.1 LBK Commands — LBK commands observe the following format:

command qualifier(/...)

Commands specify the kind of test that will be performed; the qualifiers establish specific test parameters, such as which line is to be tested and how many times the test is to be run. All commands and qualifiers can be abbreviated to three characters. The SHOW_DATA and SHOW_ERRORS qualifiers can be abbreviated as follows:

SHOW_DAT, SHOW_ERR

Commands are:

LOOP
SHOW
MONITOR
HELP

The tests that you can perform with each command in combination with its relevant qualifier(s) are described in the remaining part of this section. Section 8.4.4 lists errors that may be reported in association with each command.

LOOP. This command starts a loop test on the line requested. The loopback test transmits a data pattern on the specified line and expects to receive it back intact. The received data is compared against the transmitted pattern and any errors are reported. Consistent data errors indicate hardware problems. The is specified by the following qualifier:

/LINE=*n*

where

n is the number of the Trace unit for the line interface. If LINE=*n* is not specified, the value defaults to 0 and the line defaults to TU0. Remember that LBK does not support KBU lines.

You can specify the number of times (1. to 255.) the LOOP test is to run by means of the following qualifier:

/ITERATIONS=*n*

where

n is a number between 1. and 255.

If ITERATIONS is not specified, *n* defaults to 3.

You can specify the length of the sample message in bytes that LBK will put on the line by means of the following qualifier:

/LENGTH=*n*

where

n is a number between 3. and 255.

If LENGTH is not specified, *n* defaults to 80. bytes

You can display the data as it was sent and as it was received back by entering the following qualifier. Note that the underscore is mandatory.

/SHOW_DATA

If you want to display the data as it was sent and as it was received in the event of errors only, enter the following qualifier. Note that the underscore is mandatory.

/SHOW_ERRORS

If you do not want the CRC generated character to be sent and returned, enter the following qualifier:

/NOCRC

This switch allows you to check the data transmission independently of the CRC generation hardware or software. In this way you can determine if a problem lies in the CRC generation device. For example, if you retest a line using the /NOCRC switch and the problem disappears, your problem may be in the CRC Generator.

SHOW. This command causes the status of the following modem control signals for the specified line to be displayed:

DATA SET READY (DSR)
DATA TERMINAL READY (DTR)
REQUEST TO SEND (RTS)
CLEAR TO SEND (CTS)

You specify the line by means of the following qualifier:

/LINE=*n*

If LINE is not specified, *n* defaults to 0 and the line defaults to TU0:

You can specify the number of times the status of the modem control signal is to be displayed by means of the following qualifier:

/ITERATIONS=*n*

where

n is a number between 1. and 255.

MONITOR. This command activates the line monitor for the trace unit specified. Ensure that the line and modem are not 'looped back'. Incoming protocol messages from the host are monitored. All user application program activity must have been stopped before the LBK MONITOR command is issued.

NOTE

Only general polls for the line specified will be monitored.

If no general polls are received within 1 minute the error message 'monitor line timeout' is returned.

You can specify the line to be monitored by means of the LINE qualifier:

/LINE=*n*

where

n is a number between 1 and 255, of the trace unit on the line to be monitored.

If LINE is not specified, n defaults to 0 , so that TU0: is monitored. For each BISYNC message received, one line is printed on the terminal.

The /ITERATIONS qualifier specifies the number of lines to be printed before exiting the MONITOR mode. (Default=3.)

NOTE

In order not to lose data, the terminal being used must operate at a minimum of twice the speed of the line being monitored.

HELP. This command displays a summary of LBK command information, as illustrated in Figure 8-7.

```
LBK>HELP
 RSX 3271 PE Loopback test program
 Commands:
         LOOP      Transmits a test pattern and expect to receive
                   it back. Cable must be looped back.
                   Qualifiers: /ITE/LIN/LEN/SHOW_ERR/SHOW_DAT/NOCRC
         MONITOR   Prints out received messages from host
                   Qualifiers: /ITE/LIN
         SHOW      Shows current modem control status
                   Qualifiers: /ITE/LIN

 Qualifiers:
          /ITERATIONS=n   Number of iterations to test     [default=3]
          /LINE=n         TU Line number to test           [default=0]
          /LENGTH=n       Length of test pattern           [default=80]
          /SHOW_ERRORS    Show data buffers on error
          /SHOW_DATA      Show data buffers on transfer
          /NOCRC          Do not send or check CRC

 LBK>^Z
 >
```

Figure 8-7: Display Generated by HELP Command

8.2.4.2 Sample LBK Session — Figure 8-8 illustrates a sample LBK session, in which the LOOP, SHOW, and MONITOR commands were specified. User input is in lowercase letters. Interpretive notes are interspersed in the listing.

```
>lbk
LBK>loop line=1 ite =3 length=80
  Loopback test started
  Normal Completion
  Normal Completion
  1 Compare Errors detected in data during a 80 byte transfer
  Test Complete
```

NOTE

Three iterations were specified. Two completed normally. The third had an error.

```
LBK>loop/show_errors
  Loopback test started
  Normal Completion
  Normal Completion
  80 Compare Errors detected in data during a 80 byte transfer
  Requested buffer size=80 Transmit Count=80 Receive Count=6
```

NOTE

Default values were taken for LINE and for ITERATIONS. The first two tests completed normally; the third encountered 80 errors. Because SHOW_ERRORS was specified, the following prints out. (Only a portion of the display is shown here.) Note that the T row always shows the message that LBK places on the line; the R row shows the message that was looped back.

```
    Bytes 0-19
T 02 C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D4 D5 D6 D7 D8 D9 E2
R 40 40 22 22 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    Bytes 20-39
T E3 E4 E5 E6 E7 E8 E9 F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 C1 C2 ...
R 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...

LBK>show
   Modem status
   DSR= ON, DTR= ON, RTS= ON, CTS= OFF
   DSR= ON, DTR= ON, RTS= ON, CTS= OFF
   DSR= ON, DTR= ON, RTS= ON, CTS= ON
   Test complete
```

NOTE

Default values were taken for LINE and for ITERATIONS.

Figure 8-8: Sample Loopback Test Listing

```
LBK>monitor/ite=5
 Monitor mode
R 40407F7F2D

R 40407F7F2D

R 40407F7F2D

R 40407F7F2D

R 40407F7F2D

 Test Complete
```

NOTE

The MONITOR command was specified for execution five times. In this example, TU0: is receiving a general poll. Referring to Table 3-3, you can verify that 40 (the control unit address known to the host) corresponds to controller address STA0, which is TU0:.

```
LBK>^Z
>
```

NOTE

To exit from LBK, the user presses CTRL/Z.

Figure 8-8 (Cont): Sample Loopback Text Listing

8.2.4.3 How to Implement Your Own Loopback Test Program — This section describes the QIO calls that must be made by an application program designed to perform the loopback test functions. To implement your own program, you must have chosen the LBK and TRA options during PEGEN.

Attach. To activate the loopback test feature, the Trace unit of the line to be tested must be dedicated for exclusive use. To do this, you must use a special attach function code, IO.ATT!2. If any other application sessions (that is, TU devices) are operating on the line to be tested, the attach fails (see the error codes below). Once the attach completes successfully, the line is in loopback mode. Once in loopback mode, the line stops answering polls and is unavailable for normal operation. To perform the loopback test, the hardware must be configured appropriately, as described in Section 8.2.4. When the line is brought back into operational status with the host, you can monitor the data that is transmitted by the host.

The format for this QIO call is:

QIO$ IO.ATT!2,*lun*...

(See the *RSX-11/3271 Protocol Emulator Programmer's Guide* for details on the QIO call format.)

The error returns and their meanings are:

IE.OFL: The TU device is offline. That is, the PE is loaded but the line is not turned to ON state. Use the NCP SET LINE *line-id* ON command.

IE.RSU: Another application program has attached a TU device on the same line. Stop the other application program and retry the loopback program.

IE.PRI An attempt has been made to attach a TU device that is not a Trace unit. The TU device to be used for loopback functions must be a Trace unit. The user must respond with the correct Trace unit. This error return may also mean that the TU device was not loaded or the loopback feature was not generated in the PE. Both the LBK and TRA options must be selected during PEGEN.

Read. When the read function (IO.RLB and IO.RVB) is made to the Trace unit, data is received from the device driver (BU or DV). The data is moved from the Large Data Buffer (LDB) to the user buffer without being translated and with all the control characters intact. Only the following control characters should be used in the test message: STX, ETX, and PAD. (A PAD character is sent as the last character in a buffer and causes a transmit to complete. It is coded as octal 377.) Messages are considered complete when either an ETX or PAD character is received. If a message ends with an ETX, the CRC is checked; if the message ends with a PAD, the CRC is not checked. A CRC error may be overwritten with a data overrun (IE.DAO) error return if the user buffer is too small to receive the entire message.

When performing a read on a TRACE unit, the PE does not buffer the read in a queue as in normal read operations. If a read is not outstanding when data is received, the data is discarded. If no data is available when the application program issues the read, the read is held until either the data is received or the loopback session is terminated by a detach or kill QIO (described below). Note that there is no timeout mechanism on the read.

The normal read function can be modified to read the modem's status as well as to read data received on the line. The read modem function is enabled by setting the third QIO parameter (specifying the read type). The user must provide a word-aligned buffer of at least 4. bytes to receive the modem sense data. The first word contains the device indication and determines the values in the second word:

- If 1, the second word is CSR0 from a DU11, DUP11, or DUV11.
- If 0, the second word is the Line Status Register associated with DM-11BB from the DV-11.

Table 8-1 indicates which bits must be set in the CSR register to read the modem status of the Data Terminal Ready, Request to Send, Data Set Ready and Clear to Send indicators. Refer to the *Terminals and Communications Handbook* for bit settings corresponding to other modem indicators.

Table 8-1: CSR Bit Settings for Modem Status Reads (DU11, DUP11, DUV11 line interfaces)

Indicator	Bit
Data Terminal Ready	01
Request to Send	02
Data Set Ready	09
Clear to Send	13

Table 8-2 indicates which bits must be set in the Line Status Register for the Data Terminal Ready, Request to Send, Data Set Ready and Clear to Send indicators:

Table 8-2: LSR Bit Settings for Modem Status Reads (DV line interfaces)

Indicator	Bit
Data Terminal Ready	01
Request to Send	02
Data Set Ready	04
Clear to Send	05

The format of the read QIO call is the same as for normal reads. The error returns and their meanings are

IE.BCC: A BCC error has been detected: the CRC check character calculated by the receiver does not match the transmitted check character.

IE.TMO: A timeout has occurred, meaning that the read function did not complete within the three second limit.

IE.ABO: Operation aborted. This occurs when other errors arise, when the read operation has been killed, or when a detach abort has been made.

Write. The write function is implemented differently from the normal application write. The message to be written must be properly formatted by the user, beginning with an STX and ending with an ETX. No check is made on the format of the message. The message body should not contain PAD or SYN characters. It should contain only valid EBCDIC characters. The length of writes must be less than the length of a Large Data Buffer or, on non-UMR mapped systems, less than the maximum message size. Writes that are not formatted properly (STX-data-ETX) are transmitted until the byte count exceeds the limits mentioned in the previous sentence.

The format for the write QIO is the same as for normal writes. Writes will complete with a normal success code unless the write function times out, or a kill or detach abort has occurred. The error returns and their meanings are

IE.TMO A timeout has occurred, meaning that the write did not complete within a three second time limit.

IE.ABO A kill or detach abort has occurred.

Detach and Kill. These two operations perform the same functions in loopback mode as they do in normal applications, except that the TU devices are returned to an online condition available for use by the application program.

8.2.5 Using the Network Dump Analyzer (NDA) to Analyze Crash Dumps

If you select the complete PE system during PEGEN, the NDA utility is provided for crash dump analysis support. (You must have also selected crash dump analysis support during RSX system generation.)

NDA helps you establish the cause of system or network crashes. It compiles and analyzes information from the RSX system's executive symbol table file, the Network Symbol Table file, and the crashed system image. The information is then sent to a listing file or at a line printer in a useful format.

NDA commands can include analysis control switches that provide such options as:

- Dumping the contents of a specified octal address range of memory.
- Listing general system data (time of crash, registers in use, and so on).
- Listing general Communication Executive data. This includes the current PE process, and the current state of buffer usage.

Table 8-4 lists all the analysis control switches.

NDA command can also include function control switches that provide such option as:

- Identifying the starting block number of the secondary storage device.
- Terminating the analysis after a specified number of errors have been encountered.
- Limiting the number of pages of analysis output.
- Creating a Save file and transferring a specified amount of the crash dump to it.

You can type NDA commands at a terminal or place them in an indirect command file.

8.2.5.1 Obtaining a Crash Dump — To obtain a crash dump, control of the processor must be transferred to the Executive Crash Dump Routine following a system crash. How this control is transferred depends on the way in which the crash occurred and whether or not you have built the Executive Debugging Tool (XDT) into your system at SYSGEN.

System crashes occur in three ways:

1. The processor encounters an unintentional halt instruction (000000).
2. An infinite loop condition arises.
3. The processor encounters a program condition that causes it to trap.

When a system crash is the result of an unintentional HALT instruction or an infinite loop condition, you must restart the processor at location 40.

When a program condition causes a processor trap, and XDT is included in your system, control transfers automatically to XDT. You can then type X at your terminal and XDT will transfer control to the Executive crash dump routine. Refer to the *RSX-11M Guide to Writing an I/O Driver* for a description of XDT.

When XDT is not included in your system, a processor trap causes control to be transferred to the crash dump routine.

Regardless of the manner in which control is transferred, once the Executive Crash Dump Routine is entered, it prints the following message on the crash notification device which was specified during SYSGEN.

CRASH — CONT WITH SCRATCH MEDIA ON *xxx*

where

xxx is the mnemonic for the device which is to receive the crashed system image.

You can then put the secondary crash dump device on-line and press the CONT switch on the CPU console. The Executive crash dump routine dumps memory to the crash dump device and halts the processor upon completion.

8.2.5.2 Running NDA — You can run NDA as either an installed or an uninstalled task.

Installed Task. When you run NDA as an installed task, use the following format to return control to:

1. The Executive upon completion of the analysis

)NDA command string

2. NDA upon completion of the analysis

)NDA
   NDA) command string

Uninstalled Task. When you run NDA as an uninstalled task, control is always returned to NDA upon completion of the analysis. The commands for running NDA are:

RUN $NDA NDA) command string	which requires that NDA.TSK be present in the User File Directory corresponding to the system UIC on device LB:
RUN NDA NDA) command string	which requires that NDA.TSK be present in the User File Directory corresponding to the current UIC on device SY:

8.2.5.3 NDA Command Strings — You must use the following format to input commands to NDA:

[*Analysis*[/*sw1*]],[*Save*[/MEMSIZ:*n*]]=[*Symbols*/STB] ,[*Input*[/*sw2*]]

where

Analysis	is the Crash Dump Analysis listing file. If a file extension is not specified, the default extension will be .LST. If an analysis file is not specified, no analysis list file will be created.
[/*sw1*]	consists of the /EXIT:*n*, /LIMIT:*n*, /-SP Function Control switches for the analysis file. A description of each and their default values is provided in Table 8-5.

Save is the Crash Dump Save file. This file is a copy of the binary data that was written to the crash dump device by the crash dump routine. It allows you to create a historical record of crash dumps.

[/MEMSIZ:*n*] A description of the MEMSIZ:*n* Function Control switch and its default value is provided in Table 8-5.

Symbols/STB is the Symbol Table file for the crashed system. This file must define all Executive and Network Symbols. A description of /STB and its default value is provided in Table 8-3.

Input is the binary input to NDA. This specification can be either a device name (your secondary storage device), or the name of a Crash Dump Save file.

[/*sw2*] consists of all the Analysis Control switches and three Functional Control switches — /KMR, /SYM, and /BL:*n*. A description of the Analysis Control switches is provided in Section 8.2.5.5. A description of the /KMR and /BL:*n* switches and their default values is provided in Table 8-5.

Output specifications (to the left of the equal sign) are position dependent. Therefore, when you include both output specifications, you must place them in the command string in the positions shown. If you do not enter a specification for the Analysis file, you must place a comma before the Save file specification. The comma can appear only if there is a Save file specified.

Input specifications (to the right of the equal sign) are position independent and can appear in either order.

Any of the specifications in the command string can appear in the complete FILES-11 format — that is, complete with device, UIC, filename, file type and version number.

When you omit any of the command specifications, NDA uses the defaults shown in Table 8-3.

Table 8-3: NDA File Default Values

File	Device	User File Directory (UFD)	Default File Name	Value Type
Analysis file	SY:	current	none	.LST
Save file	SY:	current	none	.CDA
Symbols file	SY:	current	none	.STB
Input file	SY:	current	none	.CDA

The following examples illustrate the NDA file default values. Assume that the user in these examples is logged in under UIC [301,356] and that the secondary storage device is DT1:. Also assume that NDA is being operated as an installed task.

Example 1.

```
>NDA
NDA> 18MAY81,18MAY81=CEX.STB/STB,DT1:/ALL
```

This command string will create:

1. The Analysis file, 18MAY81.LST

2. The Save file, 18MAY81.CDA

The binary input is read from DECtape Unit 1 and NDA analyzes it in accordance with the information contained in the symbol definition file CEX.STB under UFD [301,356].

Example 2.

```
>NDA
NDA>,17AUG81=DT1:
```

This command string creates a Save file, 17AUG81.CDA. In this case, NDA performs no analysis, since there is no analysis listing file.

The binary input is read from DECtape Unit 1.

Example 3.

```
>NDA
NDA>=17AUG81/ALL
```

This command string produces an analysis output listing using the /ALL analysis switch (see Table 8-4 for a complete description of this switch). The binary input is read from the previously created Save file (17AUG81.CDA). NDA analyzes it in accordance with the information contained in the symbol definition file CEX.STB under UFD [301,356].

8.2.5.4 Indirect Command Files — You can enter command strings to NDA directly from the keyboard or indirectly through the use of the indirect command file facility. NDA indirect command files must not contain a reference to another command file.

8.2.5.5 Switches — Switches provide you with the means to control NDA operation. There are two types: Analysis Control switches and Functional Control switches. Analysis Control switches determine which NDA analysis routines will be applied to the input file. Functional Control switches provide a number of NDA control options. For example, such switches might terminate an analysis after NDA has encountered a specified number of errors. Or, a switch might limit the number of pages of output listing.

Analysis Control Switches. More than one Analysis Control switch may be specified. If none are specified, the default is /SYS/CEX. Table 8-4 provides a list of the Analysis Control switches and a brief description of their effect.

Table 8-4: NDA Analysis Control Switches

Switch	Description
/DUMP:*a*:*b* /DMP:*a*:*b*	Lists the contents of physical memory from octal address *a* through octal address *b*. Both *a* and *b* are 32-bit addresses.
/SYS	Lists general system data. This includes the time of the crash, the current task, and the registers in use.
/CEX	Lists general Communication Executive data. This includes the node name and number, the current PE process, and the current state of buffer usage.
/SLT	Prints interpreted contents of the System Line Tables.
/PDV	Prints interpreted contents of the Process Descriptor Vectors.
/QUE	Lists the contents of the fork process queue. /QUEUE
/LIS /LIST /LISTS /FREE	Scans the CCB and RDB free lists for errors and prints the results.
/CTB	Prints information on all Controller Blocks assigned to PE system devices (printed only on RSX-11M-PLUS systems).
/PROC /PRO	Dumps the line tables for all loaded PE processes.
/TASK /TSK	Prints the network mailbox, window blocks, and I/O packets for each task using the network.
/ALL	Executes the following switches: /SYS /PDV /PRO /CEX /QUE /PRO /SLT /LIS /TSK /CTB

Functional Control Switches. Table 8-5 provides a list of the Function Control switches and a brief description of their effect.

More than one Functional Control switch may be specified. If switches are not specified, then the default value listed in Table 8-5 is assumed. Note that when a switch requires a numeric argument, the argument is considered to be octal, unless followed by a decimal point.

Table 8-5: NDA Functional Control Switches

Switch	File	Description	Default
/BL:*n*	Input File	Identifies the starting block number of the secondary storage device. The value of *n* must be less than 65535. NDA reads the dump from the input device beginning at logical block *n*. If the crash dump device is not a disk or DECtape, NDA ignores this switch.	/BL:1
/EXIT:*n*	Analysis File	Terminates analysis after encountering *n* analysis errors. NDA maintains an error count. If you specify the /EXIT:n switch, NDA terminates analysis after *n* errors. If you specify the /EXIT but do not specify *n*, NDA exits after one error.	/-EXIT
/LIMIT:*n*	Analysis File	The /LIMIT:*n* switch limits the number of pages of analysis output. When NDA has generated *n* pages, it terminates the analysis and prints a message on the user terminal, indicating that it has done so.	/LIMIT:500
/MEMSIZ:*n*	Save File	This switch is used to create a Save File 4x*n* blocks long and to transfer *n*K words to it from the input file. The legal range of values for *n* is from 16. to 4096. This switch is only necessary when saving dumps from secondary storage devices that have no EOFs (Disk or DECtape). When the crash input resides on a magtape, the save file will be zero filled if the EOF is read before *n*K words have been transferred.	/MEMSIZ:124

Table 8-5:(Cont) NDA Functional Control Switches

Switch	File	Description	Default
/KMR	Input File	On mapped systems, when NDA reads incorrect Page Address Register (PAR) values from the crashstack, it prints an error message on the user's terminal, and attempts to continue with the analysis. In this event, you can use the /KMR switch when you restart the analysis. When you specify /KMR, NDA is forced to use standard mapping values to convert kernel virtual addresses to physical memory addresses.	/-KMR for mapped systems /KMR for unmapped systems
/-SP	Analysis File	NDA does not spool the analysis output listing file to the line printer.	/SP
/STB	Symbols File	The /STB switch identifies the file that contains both the Executive and Network Symbol Tables, for the crashed system. NDA opens the specified file and extracts the necessary symbol values. If it fails to find any required symbol values, NDA aborts the analysis and prints an error message on the terminal. If no symbol file is specified via the /STB switch, CEX.STB under the current UIC is used. **NOTE** If no symbol table file is available which contains both the Executive and Network symbols, use the Task Builder to create such a file. For example, `>TKB,,CEX=RSX11M.STB,CEX.STB`	CEX.STB in the current UIC.
/SYM	Input File	If any required symbols are not found in the symbol table file, set their values to zero and proceed with the analysis. Normally, this condition would cause NDA to abort the analysis.	/-SYM

8.3 Troubleshooting Procedures

The following subsections discuss steps to take for isolating and solving some of the common communications software and hardware problems that may arise with use of the 3271 PE. The type of problems addressed here are transmission failures that occur after the system has been tested successfully. The symptoms that appear with each of these problems are also noted. A typical problem symptom may be an unusually large number of NAKs or timeouts, or no response from the host system.

To facilitate troubleshooting and debugging efforts, it is highly recommended that you make the following provisions:

1. Use the Trace program to obtain trace listings of all communication line traffic. (You must have selected the Trace program option during PEGEN.)

2. Keep a copy of your PEGEN listing. This listing provides a record of the options and parameter specifications made during PEGEN. It gives you a clear picture of your PE configuration.

3. Generate the NDA utility so that you may have crash dump analysis support.

8.3.1 Troubleshooting Software Problems

Take the following steps, in the order given, to determine the cause of 3271 PE software communications failures:

1. **Incorrect Log-On.** If you have a dial-up line and the host system requires log-on, make sure the host operator logged on properly. Some symptoms of a log-on error are:

   a. Your PE system would not be polled by the host. You can determine if the PE is being polled by running the DCE utility or the Trace program. The DCE error counter report includes a flag that designates whether the host is active. The Trace program reports all activity on the line.

   b. Your PE system would not receive messages (check the Data Blocks Received counter in the NCP SHOW LINE COUNTER display or the DCE statistics report).

2. **Incorrect Address Selection.** Make certain that you have made proper address selection for your control units and PE station (see Section 3.5). With improper address selecton, a particular control unit or station (pseudodevice) would not receive a poll from the host. There would be no activity on the affected line(s). Check the Data Blocks Received counter.

3. **User Application Program Not Running.** For successful communications to occur, cooperating user application program (UAPs) must run at both ends of the line. The following symptoms may arise:

   a. If the host system does not have a UAP running, the PE system may eventually time out and the host may become inactive. This condition, however, is not a *hard error*. That is, indication of the condition is not necessary. An application program can be made aware of this condition by timing requests made to the PE. If the program has to wait beyond a specified time limit, it can respond with notification to the user.

   b. If no RSX UAP is running, the host system may receive an error message indicating that a particular station is not functioning. Check that the RSX UAP corresponding to the designated station is attached to it (the *RSX-11/3271 PE Programmer's Guide* discusses QIO Macro user interfaces in Chapter 2).

The *RSX-11/3271 PE Programmer's Guide* offers more tips for troubleshooting and debugging software problems.

8.3.2 Troubleshooting Hardware Problems

Take the following steps to diagnose and resolve common hardware problems:

1. **Modem Options.** Check that your modem is properly strapped for your communications circuit and applications. Options depend on these and other issues:

   a. Circuit type (for example, full- or half-duplex, dial-up)

   b. Modem equipment (line speed)

   c. Common carrier (for example, Bell, ITT, or MCI)

   d. Communications interfaces (of the RSX and the host systems)

   e. Clock source (your modems must have internal clocking)

   Improper Selection of Modem Options may be indicated by any of the following symptoms:

   a. No operation

   b. High NAK count

   c. High receive or transmission time-out count

8.3 Troubleshooting Procedures

The following subsections discuss steps to take for isolating and solving some of the common communications software and hardware problems that may arise with use of the 3271 PE. The type of problems addressed here are transmission failures that occur after the system has been tested successfully. The symptoms that appear with each of these problems are also noted. A typical problem symptom may be an unusually large number of NAKs or timeouts, or no response from the host system.

To facilitate troubleshooting and debugging efforts, it is highly recommended that you make the following provisions:

1. Use the Trace program to obtain trace listings of all communication line traffic. (You must have selected the Trace program option during PEGEN.)

2. Keep a copy of your PEGEN listing. This listing provides a record of the options and parameter specifications made during PEGEN. It gives you a clear picture of your PE configuration.

3. Generate the NDA utility so that you may have crash dump analysis support.

8.3.1 Troubleshooting Software Problems

Take the following steps, in the order given, to determine the cause of 3271 PE software communications failures:

1. **Incorrect Log-On.** If you have a dial-up line and the host system requires log-on, make sure the host operator logged on properly. Some symptoms of a log-on error are:

   a. Your PE system would not be polled by the host. You can determine if the PE is being polled by running the DCE utility or the Trace program. The DCE error counter report includes a flag that designates whether the host is active. The Trace program reports all activity on the line.

   b. Your PE system would not receive messages (check the Data Blocks Received counter in the NCP SHOW LINE COUNTER display or the DCE statistics report).

2. **Incorrect Address Selection.** Make certain that you have made proper address selection for your control units and PE station (see Section 3.5). With improper address selecton, a particular control unit or station (pseudodevice) would not receive a poll from the host. There would be no activity on the affected line(s). Check the Data Blocks Received counter.

3. **User Application Program Not Running.** For successful communications to occur, cooperating user application program (UAPs) must run at both ends of the line. The following symptoms may arise:

   a. If the host system does not have a UAP running, the PE system may eventually time out and the host may become inactive. This condition, however, is not a *hard error*. That is, indication of the condition is not necessary. An application program can be made aware of this condition by timing requests made to the PE. If the program has to wait beyond a specified time limit, it can respond with notification to the user.

   b. If no RSX UAP is running, the host system may receive an error message indicating that a particular station is not functioning. Check that the RSX UAP corresponding to the designated station is attached to it (the *RSX-11/3271 PE Programmer's Guide* discusses QIO Macro user interfaces in Chapter 2).

The *RSX-11/3271 PE Programmer's Guide* offers more tips for troubleshooting and debugging software problems.

8.3.2 Troubleshooting Hardware Problems

Take the following steps to diagnose and resolve common hardware problems:

1. **Modem Options.** Check that your modem is properly strapped for your communications circuit and applications. Options depend on these and other issues:

   a. Circuit type (for example, full- or half-duplex, dial-up)

   b. Modem equipment (line speed)

   c. Common carrier (for example, Bell, ITT, or MCI)

   d. Communications interfaces (of the RSX and the host systems)

   e. Clock source (your modems must have internal clocking)

   Improper Selection of Modem Options may be indicated by any of the following symptoms:

   a. No operation

   b. High NAK count

   c. High receive or transmission time-out count

d. Low throughput

e. Large number of CRC errors, detected by the PE and reflected in high NAK counts.

The Line Loopback Test can help isolate some of these problems.

2. **Interface Device or Cable Problem.** A DUP11, DU11, DUV11, or DV11 device may be malfunctioning, or the communications cable may be faulty. This problem may be indicated by any of the following conditions, especially the first two:

   a. No operation

   b. High NAK count

   c. High receive or transmission time-out count

   d. Low throughput

   e. Large number of CRC errors (due to line interference), detected by the PE and reflected in high NAK counts.

   Use the Line Loopback utility to test the modems and cables.

3. **Multi-Drop Line Problem.** One of the stations on a multi-drop line may fail and cause malfunctioning. In this case, a station may be hanging on a line. That is, the station is sending a non-stop stream that ties up the line. Such a problem can show up in a number of ways. Use of a datascope may be helpful for locating the problem, or you may check directly with the host.

   Symptoms of this problem may be any of the following:

   a. No operation

   b. Abnormally low receive or transmit message counts

   c. Abnormally high NAK or time-out counts, indicated by high NAK counts

   d. CRC errors due to interference

   The cause of this problem may be a bad line or circuit.

8.4 3271 Protocol Emulator Component Error Messages

This section lists the error messages generated by the following 3271 PE components.

- CFE (8.4.1)
- DCE (8.4.2)
- DLX (8.4.3)
- LBK (8.4.4)
- NCP (8.4.5)
- NDA (8.4.6)
- NTL (8.4.7)
- TRA (8.4.8)
- VNP (8.4.9)

8.4.1 CFE Error Messages

This section lists the CFE error messages and explains why they are generated.

CFE — Buffer allocation failure
CFE tried to allocate an internal buffer to complete the command that failed. If you must have a new entry in CETAB, then on a mapped system, you must remove CFE and reinstall it with an increment.

CFE — Command line error
The command line has a bad character or equivalent. Retype the command.

CFE — Consistency error
There is an internal consistency error in CFE. Contact your local DIGITAL office for information about the services DIGITAL offers to assist you in problem resolution. If you are under warranty or have purchased the 3271 PE software from DIGITAL, see the *Release Notes* for information on how to file an SPR and what information you should include with the SPR.

CFE — Error in opening file
CFE could not open the CETAB source file that was specified. Make sure that the file is where you specified, that the file is not locked and that you have write privileges to the file.

CFE — Error writing file
CFE was unable to write a record to the output file, therefore the CETAB source file was not updated. Check to make sure that the disk is not full.

CFE — Executor node not allowed with command
The command parameters that were specified in the command line are not permitted on the executor node.

CFE — Illegal command
The command line that you gave to CFE is incorrect. Type in the correct command.

CFE — Illegal device priority
The device priority specified in the command line is out of range.

CFE — Insufficient buffer space
CFE was not built with enough buffer space to accomodate the CETAB source file information. On a mapped system, you can remove CFE and re-install it with an increment.

CFE — Invalid line identification
The line identification specifed in the command line is not correct. Type in the correct line identification.

CFE — Invalid number
The command line has a number out of range. Retype the line with a valid number.

CFE — Invalid process identification
The process name specified in the command line is too long. Type in the correct process name.

CFE — Line not in configuration
The line identification that was specified in the command line is not in the CETAB source file.

CFE — Logging monitor not supported
The status of the logging monitor may not be altered since it is not supported in your configuration.

CFE — No buffer definition
CFE could not find a description of your buffers in the CETAB source file, BUF$DF macro.

CFE — No help available
CFE could not find the help file.

CFE — No node definition
The CETAB source file does not contain a definition of the local node, NOD$DF macro.

CFE — No NSP features macro
The CETAB source file does not contain a definition of the NSP features, FEA$DF macro. You will not be able to modify any of the states contained in this definition.

CFE — No unit CSR on this line
The line that you attempted to change the unit CSR for, does not have a unit CSR. This part of the command line was ignored.

CFE — Process not in configuration
The process name that was specified in the command line is not in the CETAB source file.

CFE — Syntax error
CFE failed to parse a line read from the CETAB source file. The bad line of the source file is displayed. This error is probably caused by someone wrongly editing CETAB. Restore the line to its original content. If the line was not modified, contact your local DIGITAL office. If you are under warranty or have purchased the 3271 PE software from DIGITAL, see the *Release Notes* for information on how to file an SPR and what information you should include with the SPR.

CFE — Unknown help qualifier
CFE has no help on the qualifier requested.

8.4.2 DCE Error Messages

The statistics log program (DCE) error messages are:

DCE — Assign *lun nnnnnn* DSW for line *x*
This error message appears when it is impossible for the Trace program to assign the *lun* for this line. *nnnnnn* is the Directive Status Word (DSW) error code which applies. The most common errors occur if the 3271 PE is not loaded or the DCE program was generated for another system. *x* is the line number that was to be logged.

DCE -- Attach I/O error IOSB: *nnnnnn llllll* line *x*
This error occurs when the attach QIO function is used for the line that is to be read for the statistics table. The two words of the I/O Status Block are formatted. Look at *nnnnnn* for the reason code associated with the failure. *llllll* is the number of bytes transferred and should always be 000000 for this message. *x* is the line number of the operation.

DCE — Read I/O error IOSB: *nnnnnn llllll* line *x*
This error occurs if there is a read problem on the unit where the statistics for the line are to be read. *nnnnnn* is the first word of the I/O Status Block for the line to be read, *llllll* is the number of bytes transferred, and *x* is the line number.

DCE — Detach I/O error IOSB: *nnnnnn llllll* line *x*
This error occurs if there is a problem detaching the TU device that was used to read the statistics for line *x*. *nnnnnn* is the completion code from the I/O Status Block. *llllll* is the count for the detach. It should always be 000000.

DCE — File system error *nnn* at *aaaaaa*

This error occurs if there is a file system error in writing the statistics file. *nnn* is the file system error code in octal. *aaaaaa* is the address in the program where the error occurred. It is probably of no value to the user, but it can be of use to software support.

8.4.3 DLX Error Messages and Codes

This section lists the DLX Error Codes. (DLX is included with all RSX-11M-PLUS systems.)

Code	Octal	Decimal	Explanation
IE.ALN	177736	-34.	The lun you specified while attempting to open a line is already in use.
IE.NLN	177733	-37.	No line has been opened with the lun you specified
IE.ABO	177761	-15.	This error is returned for all I/O operations after a link fails and until the link is reinitialized.
IE.DNR	177775	-3.	The hardware device was not ready. The line was stopped and has not been restarted.
IE.SPC	177772	-6.	The buffer you specified is too large. (PDP-11/70 with extended memory only.)
IE.TMO	177666	-74.	The receive timer expired before a data message was received.
IE.DAO	177763	-13.	Either a message was lost or the user buffer was too small to receive the data.
IE.PRI	177760	-16.	The line is not available for use by DLX.
IE.RSU	177757	-17.	The line that you specified is already in use.
IE.ONP	177773	-5.	The QIO directive specified an illegal subfunction code.
IE.VER	177774	-4.	A physical line error has occurred, as indicated by one of the following error codes, which appears in the second word of the status field: 100376 Receive threshold exceeded 100374 Rep threshold exceeded 100372 MOP threshold exceeded 100370 General error 100366 Disconnect 100364 Start received 100362 Operation aborted 100361 NAK threshold exceeded

8.4.4 LBK Error Messages

The loopback program can generate a number of error messages for each command. Some of the error messages indicate simple operational problems, such as command entry errors or incorrect line number specification. Other error messages are reports that indicate a problem exists within your PE configuration.

There are four categories of LBK error messages:

1. Those that can arise with the use of any command (General Errors).
2. Those specific to use of the LOOP command.
3. Those specific to use of the SHOW command.
4. Those specific to use of the MONITOR command.

All commands

The following messages may arise with use of any of the LBK commands:

Command entry error

Indicates that error occurred while trying to read the command line. Try again.

Syntax error in command

Indicates that the command given was not a proper LBK command. This could be due to misspelt keywords or incorrect values for any qualifier.

Failed to assign unit

Indicates that an error occurred while LBK tried to assign to the TU unit specified. Probably indicates that the TUD is not loaded or the unit number does not exist. The reason code is given in the Directive Error message, following. The Directive Error message always appears with this error message.

Directive Error ⟨*octal number*⟩

Indicates that a RSX directive error occurred. The number displayed is the octal value of the $DSW word returned. See your *RSX-11M/M-PLUS Mini Reference* or *Executive Reference Manual* for the meaning of these codes.

Failed to attach unit

Indicates that an error occurred while LBK tried to attach to the TU unit specified. A second error message follows indicating the reason. If none of these errors is the problem, a general IO error message is printed.

TU line is not turned ON. The TU unit specified has not been set on by NCP. Turn the line on and try again.

TU unit is in use. The TU controller has other activity and cannot be set to loopback mode. You must stop all activity on the controller before LBK can use the line.

TU line is not a Trace line, or not loaded, or not supported. The TU line specified must be a Trace line (that is, having the lowest number on a controller); the DDM for the device interface (BU,DV) must be loaded in the system; and Trace support must have been selected during PEGEN. Note that LBK does not support KBU lines.

IO Error IOSB=⟨octal word⟩ IOSB+2=⟨octal word⟩

General QIO error return report. The I/O Status Block, words 0 and 2 are listed. See your *Programmer's Guide* or the *RSX-11M/M-PLUS I/O Driver's Manual* for the meaning of these codes.

LOOP command errors

Two types of errors may arise while loopback testing: (1) I/O errors in which an I/O code is returned on completion of the test transfer, and (2) data errors in which data received is incorrect. You must determine the cause of the problems indicated by these codes or messages.

Transmit timeout on transfer

Indicates that the PE was not able to correctly transmit the test pattern. Possible reasons are: no synchronous clock supplied from the modem or modem eliminator, incorrect cable connections, or cable too long.

Receive timeout on transfer

Indicates that the PE never received the test pattern. Possible reason is that line is not looped back.

Error return on read, IOSB=⟨octal word⟩ IOSB+2=⟨octal word⟩

Indicates an error return on the read QIO. The I/O status is displayed. LBK should only get the following error. If any others occur, consult your *Programmers Guide.*

CRC error. The CRC received was not correct. You could have problems with your CRC generating software or hardware. Try again with the /NOCRC switch.

Compare errors detected in data during ⟨number⟩ byte transfer

The data received did not match the transmitted pattern. This indicates problems with the line interface or the quality of the modem or transmission connections. The /SHOW_DATA and /SHOW_ERRORS qualifiers allow you to see the data and the discrepancies.

SHOW modem status errors

These error messages are very unlikely to occur.

Modem sense read error

Indicates an error occurred while trying to read the modem status from the PE. An I/O status block message will follow.

Unknown device type returned

Indicates that the device type return in the modem sense message was incorrect.

MONITOR mode errors

Monitor mode waits to receive messages on the TU line specified. If no data is received, you can exit from the monitor mode only by typing a CTRL/C and aborting LBK.

Monitor line read error

Indicates that an error was returned on the read from the PE. A general I/O status block message follows. These errors are not serious.

Monitor Line Timeout

Indicates that no general poll has been received within 1 minute of issuing the loop monitor command.

8.4.5 NCP Error Messages

This section lists the error messages from the full capability NCP and explains why they are generated.

The general format of an NCP error message is:

NCP — [⟨*component*⟩,]⟨*command*⟩ ⟨*diagnostic*⟩, ⟨*error message*⟩
[, ⟨*error detail*⟩] [*extra text*]

where:

⟨*component*⟩ is the specific component identification, for example LINE *line-id*. Appears only in a command affecting multiple components.

⟨*command*⟩ is the command that caused the error.

⟨*diagnostic*⟩ is one of the following:

FAILED

Command accepted by NCP, but execution of command failed.

NOT ACCEPTED

Command execution not attempted; NCP would not accept the command.

⟨*error message*⟩ is the major reason for the failure.

If for any reason NCP cannot interpret an internal error code, the error message field will contain one of:

Management return #*n*

An error code of *n* decimal was returned by the Network Management software.

System return #*n*

An error code of *n* decimal was returned by the RSX operating system software.

⟨*error detail*⟩ is a more detailed explanation of the failure for certain error messages.

If for any reason NCP cannot interpret an error detail code, the error detail field will contain

detail #*n*

meaning that an error detail of *n* decimal was returned by the Network Management software.

⟨*extra text*⟩ is an additional, system-specific explanation of the error condition.

NCP Error Messages

The following error messages are RSX system specific.

NCP — Error reading command
NCP could not read a command from either the terminal or a command file because of either a hardware error or some other unrecognized command read error condition.

NCP — Exceeded command file nesting depth
A command file, executing in NCP, attempted to start another command file when there were already six command files nested.

NCP — File privilege violation
The file specified with the TO option in a SHOW command is not available to the user of NCP.

NCP — Invalid device name syntax
The device name in a command file or TO file specification did not have legal syntax.

NCP — Invalid directory syntax
The directory identification (UIC) in a command file or TO file specification did not have legal syntax.

NCP — Invalid file name syntax
The file name in a command file or TO file specification did not have legal syntax.

NCP — NCP program error
NCP detected an internal programming error.

NCP — Unrecognized device or unit
The device name or unit number in a command file or TO file specification could not be found by the RSX operating system.

NCP — Unrecognized file, device, or directory
The file name, device name, or directory (UIC) in a command file or TO file specification could not be found by the RSX operating system.

The following error messages are standard in all implementations.

NCP — Bad management response
The information returned to NCP from either its internal Network Management functions or from the Network Management Listener was improperly formatted or contained an invalid value. This error condition implies a programming error in the module that responded to NCP.

NCP — Component in wrong state
The current operational state of the component precludes the requested operation. For example, you cannot clear a line that is turned ON.

The error detail included with this message will be the type of the component. For example, LINE, PROCESS, etc.

NCP — File I/O error
A hardware error was encountered while reading or writing one of the files necessary for the requested operation.

NCP — File open error
One of the files necessary for the requested operation could not be reserved for use.

The error detail is the same as displayed for File I/O Error.

NCP — Hardware failure
The requested operation could not be performed because of the failures of a related hardware component to function as needed.

NCP — Invalid file contents
The information found in the files necessary for the requested operation was of invalid form or value.

The error detail is the same as displayed for File I/O Error.

NCP — Invalid identification format
The identification of the component involved in the requested operation did not have proper syntax. For example, a non-alphanumeric character in a node name.

The error detail included with this message will be the type of the component. For example, LINE, PROCESS, etc.

NCP — Invalid message format
The information sent by NCP to either its internal Network Management functions or to a Network Mangement Listener was improperly formatted or contained an invalid value. This error condition implies a programming error in NCP.

NCP — Invalid parameter grouping
The parameters furnished by the user for the requested operation cannot be included in a single command.

NCP — **Invalid parameter value**
The value of a parameter furnished by the user for the requested operation was not acceptable. For example, a numeric parameter out of range.

The error detail included with this message will be the type of the parameter.

NCP — **Management program error**
The Network Management software has detected an internal programming error.

NCP — **Operation failure**
The requested operation failed.

For some RSX system-specific failures, one of the following extra text messages may be included.

DLX error *n*
A direct line control (DLX) function failed with decimal error code *n*. Consult the direct line control documentation for an explanation of the error number.

Network Initializer function failed
The Network Initializer (NTINIT) failed to perform its part of the requested operation. The network initializer will write an error message to the requesting terminal.

Network Initializer not installed
The Network Initializer could not run because it is not properly installed.

Network loader function failed
The network loader (NTL) failed to perform its part of the requested operation. The network loader will write an error message to the requesting terminal.

Network loader not installed
The network loader could not run because it is not properly installed.

NCP — **Parameter missing**
The user did not supply a parameter that is necessary for the requested operation.

The error detail included with this message will be the type of the parameter.

NCP — **Parameter not applicable**
The user supplied a parameter that is not applicable to the requested operation on the specified component.

The error detail included with this message will be the type of the parameter.

NCP — Parameter value too long
The user supplied a parameter value that was too long to be accepted by the Network Management Listener.

The error detail included with this message will be the type of the parameter.

NCP — Privilege violation
The user does not have sufficient privilege to perform the requested operation.

NCP — Resource error
Network Management had insufficient internal resources to perform the requested operation.

NCP — System-specific management function not supported
The requested operation is RSX system-specific and is not supported by the Network Management Listener.

NCP — Unrecognized command
NCP does not have the command that the user typed.

NCP — Unrecognized component
The component specified by the user does not exist.

The error detail included with this message will be the type of the component. For example, LINE or PROCESS.

NCP — Unrecognized function or option
The requested operation is not implemented by the executor.

NCP — Unrecognized keyword
One of the keywords in a command is not known by NCP.

NCP — Unrecognized parameter type
One of the parameters given by the user is not implemented by the executor.

The error detail included with this message will be the type of the parameter.

NCP — Unrecognized value
A parameter value given by the user was not known to NCP.

NCP — Value out of range
A parameter value given by the user is out of range implemented in NCP.

8.4.6 NDA Error Messages

NDA prints an error message on your terminal when it detects one of the error conditions described below. These error messages reflect operational conditions. Do not confuse these messages with the diagnostic analysis messages that NDA generates during the analysis, and prints in the analysis listing.

Note that not all of the messages listed below result in termination of the analysis. Some are informational in nature, while others are fatal errors.

NDA — Analysis output must be directed to an explicit device or file
Fatal error. NDA requires an explicit output file specification. There are no default output file names.

NDA — Analysis terminated after *n*. pages
Informational error. NDA terminated the analysis after generating *n* pages of analysis output.

NDA — Command I/O error
Fatal error. The system returned an error when NDA attempted to read a command line.

NDA — Command line syntax error
Fatal error. NDA detected an error in the syntax of an NDA command line. NDA will point to the error within the command line.

NDA — Crash dump must be input from an explicit device or file
Fatal error. The crash dump input file specification must be explicit. There is no default file specification for the crash dump input.

NDA — Device driver missing
Fatal error. You have not loaded the driver for the crash dump input device.

NDA — Errors detected: *n*.
NDA has detected *n* analysis errors during the run. Analysis errors arise when, for example, the database is damaged. A queue pointer may be pointing to an odd address.

NDA — Error reading crash dump
Fatal error. NDA either could not read or could not access the crash dump input device. The device may not be properly allocated or may contain bad blocks.

NDA — Error reading file *filename*
Fatal error. The system returned an error when NDA attempted to read the crash dump file. This could be caused by:

1. A device error
2. A bad block on the volume

NDA — Error reading symbol file *filename*
Fatal error. The system returned an error when NDA attempted to read the symbol table definition file.

NDA — Error writing analysis file
Fatal error. The system returned an error when NDA attempted to write a line into the analysis listing file. This could be caused by:

1. A problem with the device
2. A full volume
3. A bad block on the volume

NDA — Error writing dump file *filename* **ERROR=***n* **DSW=***n*.
Fatal error. The system returned an error when NDA attempted to write into the binary output file. This condition could be caused by:

1. A problem with the deivce
2. A full volume
3. A bad block on the volume

NDA — Exiting due to illegal trap — snapshot dump being attempted
Fatal error. NDA has aborted after detecting an odd address or some other type of fault. If PMD is installed in the system, the system will generate a snapshot dump. This is an indication of a hardware or software problem.

NDA — Failed to assign LUN to input device *ddu*
Fatal error. The ASSIGN LUN directive failed when NDA attempted to use it to attach the specified input device, before reading the crash dump from the device. The ASSIGN LUN will fail if the device name in the NDA command line is invalid.

NDA — Failed to extend page buffer — *n* pages available
The 'extend task' directive failed when NDA attempted to use it to expand the page buffer. This problem will cause the analysis to take longer, but the analysis will continue with a buffer of *n* pages, each 256 words long.

NDA — Failed to open input file *filename*
Fatal error. One of the following conditions exists:

1. The specified device does not exist.
2. The volume is not mounted.
3. A problem exists with the device.
4. The specified UFD does not exist.
5. The specified file does not exist.
6. You do not have read access privileges.

NDA — Failed to open output file *filename*
Fatal error. One of the following conditions exist:

1. The specified device does not exist.
2. The volume is not mounted.
3. A problem exists with the device.
4. The specified UFD does not exist.
5. The volume is full or the device is write-protected.
6. You do not have write access privilege to UFD.

NDA — Illegal switch
Fatal error. You have specified an unknown switch or have used a legal switch after the wrong file specification.

NDA — Indirect file open failure
Fatal error. NDA could not open an indirect command file specified as "@filename" in the NDA command line.

NDA — Indirect command syntax error
Fatal error. The name of the indirect command file (*@filename*) is syntactically incorrect.

NDA — Maximum indirect file depth exceeded
Fatal error. You have exceeded the maximum allowable number of nested indirect command files (only one level of indirection is permitted in NDA). The rest of the top level command file will be executed.

NDA — No input file specified
Fatal error. You did not supply an input file specification for the crash dump.

NDA — No output file specified
Fatal error. You have not specified an output file.

NDA — Output dump *filename* must be explicit
Fatal error. See preceding message.

NDA — Premature end of dump input — *filename* being zero-filled
NDA reached the end of the medium (or end of file mark if a magtape) before the crash dump output file had been completely filled. If you expected the file to be completely filled by the dump, this condition could indicate a problem.

NDA — Symbol file *filename* has illegal format
Fatal error. The specified executive symbol table file has an improper format, probably caused by entry of the wrong file name. However, this message could also indicate a problem with the device or medium on which the file is located.

NDA — Symbol *symbolname* not defined in symbol file
Fatal error. NDA did not find a symbol that it required for the analysis in the specified executive symbol table file. The analysis may be restarted specifying the /SYM switch forcing NDA to continue with the analysis, using a zero value for all undefined symbols.

NDA — Transfer complete — *ddu* may be unloaded
The transfer of the crash dump to the output file is finished; you may unload the crash dump device. This message occurs only when you have specified a binary save file in the command string to NDA.

NDA — Unknown GET command line error
Fatal error. Unrecognized error has occurred when NDA attempted to read a command line.

NDA — Warning kernel PARs do not contain proper values
This message appears only when dumping mapped systems. It indicates that the values contained in the Page Address Registers are invalid. NDA will attempt to finish the analysis using the indicated values. When restarting the analysis, you must specify the /KMR switch (Section 8.2.5.5). This switch forces NDA to use standard mapping values when converting kernel virtual addresses to physical memory addresses.

8.4.7 NTL Error Messages

The Network Loader (NTL) loads and unloads network software in response to NCP commands. This section lists the NTL error messages and explains why they are generated.

When an NCP command executes locally, NTL error messages display on the requesting terminal. When the command executes remotely, NTL error messages display on the console terminal of the remote system.

Some of the error messages include strings which are replaced with run-time variables. The strings and what they represent are:

xxx a network process or device driver name

xxx-n a controller name

xxx-n-n a line name

pppppp a partition name

vvv a vector value

dd: an RSX device name

Any strings enclosed within angle brackets may or may not appear on the actual error message depending on the context in which the error is detected and printed.

NTL — **CCB allocation failure**
There is not enough space for all of the CCBs. It is not known how many were successfully allocated.

NTL — **CETAB allocation failure**
There is not enough dynamic memory to allocate the CETAB database.

NTL — **Channel *nn* still active**
You have tried to CLEAR PROCESS of an LLC before all of its lines are OFF. Set all of the lines to OFF first.

NTL — **Comm Exec incompatible with RSX-11M system**
The Communications Executive has been built with the incorrect RSX11M.STB file.

NTL — **Config file — device *dd:* UNIBUS run mask missing**
RSX-11M-PLUS only. The UNIBUS run mask is missing for the specified device.

NTL — Config file — device *xxx-n* offline CNT$DF *ctlnum, vaddr, pgaddr, pri*
The controller's CSR address (*pgaddr*) specified in the Configuration file does not exist on the UNIBUS.

NTL — Config file — device *xxx-n* secondary CSR offline UNT$DF *unit, cha1, cha2, seccsr*
This error pertains to the KMC device only.

The secondary CSR (the CSR of the DUP device that the KMC will be driving) does not exist. Use CFE to change the secondary CSR address (*seccsr*) in the Configuration file. The 'SET LINE CSR' keyword overrides the main CSR only.

NTL — Config file — partition *pppppp* sub-PCB allocation failure PAR$DF *name1,name2,...*
There is not enough space for a partition control block for the network buffer pool.

NTL — Config file — vector *vvv* not in system CNT$DF *ctlnum, vaddr, pgaddr, pri*
The controller's vector address *vaddr* specified in the Configuration file does not exist in your system's vector space.

NTL — Configuration file — binary buffer overflow
The NTL end-of-task buffers are not large enough to hold the input. You must rebuild NTL for an unmapped system or install NTL with a /INC option for a mapped system.

NTL — Configuration file — LLC$DF process extension too big
The TUD process has an extension which will make the process size bigger than the 4K maximum allowed.

NTL — Configuration file — object table allocation failure
There is not enough dynamic memory to allocate the object table.

NTL — *dd:* Micro-code load — unable to find UNIBUS run mask (RSX-11M-PLUS only)
The UNIBUS run mask could not be found for the device which needed to have micro-code loaded.

NTL — *dd:* Device tables — allocation failure
There is not enough space for the RSX-11M device data structures (DCB, UCB, and SCB) for one of the network 'devices'. Device, in this case, does not refer to a piece of hardware, but rather to a software structure to which the user program assigns its LUNs and directs its QIOs.

NTL — ICB allocation failure
There is not enough space in DSR to allocate an ICB (interrupt control block).

NTL — **Incompatible with Comm Exec**
NTL has been built with the incorrect CEX.STB file.

NTL — **Incompatible with RSX-11M system**
NTL has been built with the incorrect RSX11M.STB file.

NTL — **KRB allocation failure**
(RSX-11M-PLUS only)
There is not enough space in either the network pool or DSR for the KRB (controller request block).

NTL — **Library block allocation failure**
There is no space in the network buffer pool.

NTL — **Line not in system**
The line you tried to SET LINE does not exist.

NTL — **Line still active**
You tried to CLEAR LINE while the line was still ON. Set the line to OFF first.

NTL — **Line *xxx-n* still ready**
You tried to CLEAR PROCESS of a DLC or DDM before all of its lines are OFF and unloaded. Set lines to OFF and unload the name line first.

NTL — **Line table allocation failure**
There is not enough space for either a device line table or an LLC process data base.

NTL — **Not enough contiguous UMRs**
Cortiguous UMRs are needed to map the network buffer pool. Issue the SET SYSTEM command before you begin fragmenting the UMRs by running your other jobs. If this fails, you can use VNP to load the network software into the operating system file instead of using NCP on the in-core image.

NTL — **Process not in system**
The process you tried to SET PROCESS does not exist.

NTL — *xxx* **Process file incompatible with Comm Exec**
The process has been built with the incorrect CEX.STB file.

NTL — *xxx* **Process file — partition** *pppppp* **sub-PCB allocation failure**
There is not enough space for a partition control block for the named process.

NTL — **Re-configuration device not in system**
(RSX-11M-PLUS only)
The re-configuration device for the network devices is not loaded.

NTL — Station not in system
There was a station found in the configuration file which was not found in the system.

NTL — Resource allocation failure
There is not enough space for the allocation size needed. (Another error message follows which contains the allocation type being attempted.)

NTL — Timer cells not found
NTL could not find the process timer cells in the process AUX.

NTL — UMR block allocation failure
There is not enough space for an UMR assignment block.

NTL — UNIBUS run not accessible
(RSX-11M-PLUS only)
The network loader could not find the CPU that the device is connected to.

NTL — Unload block allocation failure
There is no space in the network buffer pool.

NTL — Vector *vvv* not available
The indicated vector is already assigned to some other user. The vector must be deassigned from the current user.

NTL — Vector *vvv* not in system
The vector specified in the command line does not exist in your system's vector space. The upper limit of the vector space is determined during SYSGEN. Repeat both SYSGEN and PEGEN.

NTL — Vector setup error
Some other user is using one or more of the interrupt vectors. This occurred after NTL first verified that the interrupt vectors were available, but before NTL built the line table and pointed the vector at the table. The vector must be de-assigned from the other user.

NTL WARNING — MUX update only
This is just a reminder that the vector and CSR for the device specified in the command line was not changed. This is because all lines on a multiplexer device are loaded when any line on the device is loaded. Therefore all subsequent loads for other lines on this device do not change the vector or CSR.

NTL WARNING — failed to re-initialize HRC
(RSX-11M-PLUS only)
The network loader could not successfully re-initialize HRC after loading a network device. This could affect the use of the system utility ...CON.

NTL WARNING — Controller parameters not changed

This warning concerns multiplexer devices. It is a reminder that the vector and CSR for the device was not changed. This is due to the fact that all lines on a multiplexer device are loaded when any line on the device is loaded. Therefore, all subsequent loads for other lines on this device do not change the vector or CSR.

NTL Partition Errors

There are two error message formats for partition errors. The first is used for network buffer pool partition errors and includes a copy of the Configuration File source line, which contains the buffer pool parameters. (The angle brackets are not part of the error text.)

NTL — Config file — partition *pppppp* ⟨*error description*⟩ **PAR$DF** *name1,name2,...*

This format is used for partition errors for the Communications Executive, network processes, and network device drivers.

NTL — *xxx* **Process file — partition** *pppppp* ⟨*error description*⟩

Communications Executive errors cause *xxx* to be replaced by CEX, although the Communications Executive is not really a network process. This reduces error message complexity.

⟨*error description*⟩ may be one of the following:

Base address mismatch

This base address of the partition does not coincide with the base address you specified during PEGEN. Either task build the process again with the correct partition limits, or re-create the partition with the proper base address. Refer to the TKB map for the task's address limits.

Busy

The partition is occupied by either network process or an RSX-11M resident library. The current occupant must be unloaded.

Not common

NTL requires the partition to be common. The partition has been created as either task or system-controlled. Remove the partition and then re-create it with the common attribute:

MCR⟩SET /NOMAIN=*pppppp*
MCR⟩SET /MAIN=*pppppp:aaaa:bbb:*COM

Not in exec space

This error pertains to the Communications Executive partition for mapped systems only.

The Communications Executive partition must be contained completely in the first 20K words of physical memory. This error indicates that the ending address of the Communications Executive's partition is larger than this limit. The Communications Executive contains an end-of-partition buffer pool; thus, if you exceed the limit by a small amount (100 or 200 octal), you can try to reduce the Communications Executive's partition size. Otherwise, you should repeat the PEGEN.

Not in system
The partition specified does not exist. Either create a partition with the name that NTL is looking for, or rebuild the component to change the partition assignment.

Not system-controlled
This error pertains to the network buffer pool only.

NTL requires the partition to be system-controlled. This error indicates that the partition was created as task or common. Remove that partition and then re-create it with the system-controlled attribute:

```
MCR) SET /NOMAIN=pppppp
MCR) SET /MAIN=pppppp:aaa:bbb:SYS
```

Too fragmented
This error pertains to system-controlled partitions only. The size of the largest free space in the partition is smaller than the size of the process being loaded. The current occupants of the partition must be rearranged to make a free space of the size needed. NTL does not activate the RSX-11M shuffler task Therefore, manual recovery is necessary.

Too small
This error pertains to common partition only. The partition is smaller than the size of the process being loaded. The partition must be created with a larger size. Refer to the TKB map for the process to determine the partition size needed.

Too small for RDBs/SDBs
This error pertains to the network buffer pool partition only. The network buffer pool is not large enough for the large and small data buffers. You can either use CFE to reduce the buffer counts and/or sizes, or re-create the partition with a larger size. If you are running on a mapped system, you may consider using a system-controlled partition for the buffer pool, in which case, NTL calculates the size for you.

8.4.8 TRACE (TRA) Program Messages

The error messages for the 3271 PE Trace program (TRA) are given below with an explanation of why they are generated.

TRA — Bad GCL file specifier
The indirect command specifier to the Trace program was not a legal RSX-11M file name. Retype the file specifier correctly.

TRA — Open error on indirect file
The file specified to the Trace program was not on the correct device, did not have the correct UIC or was not correct. Supply the corrected information to the Trace program.

TRA — Command file error
There was an error in reading the command file. Either there was a device error or a command line exceeded 80 decimal characters in length.

TRA — Assign LUN error — nonexistent TU device
The TU device assigned as the trace input device was not in the system, the wrong TU unit was specified or the RSX-11/3271 PE was not loaded.

TRA — Attach failure on TU device
The TU device requested was already attached by another task. To continue, abort the task currently owning the TU device and retype the command string.

TRA — Error in command line syntax
There was an error in the input command line. The output file specifier was incorrect or an incorrect switch was specified. There was no = in the command string, or the input device was not a TU device.

TRA — Error in opening output file
The Trace program could not open the output file. The output device was off-line, write locked, or nonexistent. The Trace program was not privileged or did not have write access to the output file UIC.

TRA — TU*n*: TRACE complete
The Trace on unit TU*n*: has completed either because the time limit has been exceeded or the number of pages to be printed is exceeded.

8.4.9 VNP Error Messages

This section lists the VNP error messages that may arise and explains why they are generated. Messages not in this list are in the list of Network Loader (NTL) messages (Section 8.4.7).

VNP — Attempt to reference block number that is not within the system image file *nnn* BIAS:*n*ADDRESS:*m*
An attempt was made to reference block number *nnn* which is not within the system image file. This error is often caused by an attempt to top-down load in a system that is greater than 124K, or by building a system image file that is smaller than the real memory size while top-down loading. Use CFE to disable top-down loading.

VNP — CETAB data structures are inconsistent
The data structures in the Communications Executive Configuration File (CETAB) are inconsistent.

VNP — Communications Executive symbol missing from symbol table *nnnnnn*
The symbol *nnnnnn* is not in the Communications Executive symbol table.

VNP — Communications Executive load failed
VNP could not successfully load the Communications Executive.

VNP — Communications Executive symbol table file open failure *nnn filename*
The Communications Executive symbol table file, *filename*, failed to open. The variable *nnn* is an error code returned by the operating system's file services. The system image file, its symbol table, and all network files must reside under the same UIC.

VNP — Configuration file read error
VNP could not read a record from the CETAB source file. This could be due to a disk error, for example.

VNP — Controller number too large
The controller number on the line identification is illegal.

VNP — Error parsing file name — *filename*
The file name of the operating system disk image file is invalid.

VNP — Executive symbol missing f[illegible]a symbol table *nnnnnn*
The symbol *nnnnnn* is not in the Communications Executive symbol table.

VNP — [illegible]ecutive symbol table file open failure *nnn filename*
[illegible]he Communications Executive symbol table file *filename* failed to open. The variable *nnn* is an error code returned by the operating system's file services. The system image file, its symbol table, and all network files must reside under the same UIC.

VNP — Failed to close system image file *nnn*
The system image file failed to close. The variable *nnn* is an error code returned by the operating system's file services.

VNP — Failed to get command line
VNP could not read a command line.

VNP — Failed to load a process
VNP did not successfully load a process that was marked for load. You can list all processes to find out which process did not load.

VNP — Failed to load a system line
VNP did not successfully load a line that was marked for load. You can list all lines to find out which line did not load.

VNP — Function not implemented
An attempt was made to execute an unimplemented VNP function.

VNP — 'GEN' partition spans highest block allocated
The default partition, GEN, spans the highest block allocated. Copy the system image file, making it larger with the PIP/BL:*xx*.switch.

VNP — Illegal executive size *nnn*
The executive size *nnn* is illegal. This is probably caused by a corrupted system image file.

VNP — Illegal unit CSR address
The unit CSR address is invalid.

VNP — Insufficient space in template processing buffer (TMP$$A *psect*). Current size is *n* bytes, but it should be expanded to at least *m* bytes
VNP's processing buffer is *n* bytes long, but requires *m* bytes for the configuration. Edit VNPGEN.CMD and extend the buffer size. Task build VNP again.

VNP — Invalid line identification
The line identification is not in the proper format.

VNP — I/O Error accessing system image file *nnn*
A disk I/O error occurred when VNP accessed the system image file. The variable *nnn* is an error code returned by the operating system's file services.

VNP — KMC microcode load failed
VNP failed to load the KMC microcode.

VNP — Line name too long
Line names must be 1 to 3 RAD50 characters.

VNP — Line parameter not applicable
The line option is valid on 'SET NODE' requests only if the node identification is a name.

VNP — Maximum controllers limited to 64
The maximum controller count for the 'SET PROCESS' request is limited to 64.

VNP — Maximum lines limited to 64
The maximum line count for the 'SET PROCESS' request is limited to 64.

VNP — Maximum transfer size with GETRV/PUTRV macros is 512 bytes, but specified record size is *n* bytes

VNP — Network buffer pool is too small
The network buffer pool is not large enough to load the system software. You will have to VMR your system image with a larger network pool.

VNP — Network Initializer not installed
The Network Initializer, NTINIT, is not installed. Install NTINIT with VMR before restarting VNP.

VNP — Network pool allocations are arbitrarily limited to 32767. words but current request is for *n* words
This error can only be overcome by source modifications. The request is made internally. You can regenerate the PE to allocate more pool or, perhaps, decrease the size of the Communications Executive partition. To allocate more pool, you must allocate less memory for those elements that take up pool space, such as CCBs.

VNP — Network pool partition not in the system
VNP could not find the network pool partition as specified during NETGEN.

VNP — Non-UMR-mapped area is too large
The portion of the network pool that is not UMR-mapped is larger than the entire network pool.

VNP — No part of 'GEN' partition is within system image
The default partition, GEN, is not within the system image. Copy the system image, making it larger, with the PIP/BL:*xx*. command.

VNP — Not enough contiguous UMRs to map network pool
The network pool needs too many UMRs to be loaded. You should try to reduce the size of the network pool.

VNP — Open failure on configuration file
VNP could not open the CETAB source file.

VNP — Open failure on system image file *nnn filename*
The system image file *filename* cannot be opened. The variable *nnn* is an error code returned by the operating system's file services.

VNP — Owner process name too long
Process names must be 1 to 3 RAD50 characters.

VNP — Partition name too long
The partition name must be 1 to 6 RAD50 characters.

VNP — Process name too long
Process names must be 1 to 3 RAD50 characters.

VNP — RDB size has been set to *nnn*
The RDB buffer size was not large enough for the network that you had configured, therefore the RDB size has been set to *nnn*.

VNP — Resource error trying to allocate CCBs
There is not enough dynamic storage space in your system to allocate the control buffers.

VNP — Reverse mapping table inconsistent
The reverse mapping table has been corrupted.

VNP — RSX-11M system pool allocation failure
The system pool is empty. Re-VMR your system increasing the size of the system pool.

VNP — Supposed system image file not actually a system image *filename*
The indicated file is not a system image file.

VNP — Syntax error
The command line is not correct.

VNP — Syntax error in BUF$DF macro
VNP could not parse the buffer definition macro in the CETAB source file.

VNP — Syntax error in PAR$DF macro
VNP could not correctly parse the partition definition macro in the CETAB source file.

VNP — System image does not contain a 'GEN' partition
The default partition for network software is GEN. The system image file does not contain a GEN partition.

VNP — System image file corrupted because VNP was terminated abnormally *filename*
A previous execution of VNP terminated abnormally and may have corrupted the system image file *filename*. Recopy the system image file before restarting VNP.

VNP — System image must be extended to *n* blocks if the 'GEN' partition is to be contained within the image
The default partition, GEN, is not within the system image. Copy the system image, making it larger, with the PIP /BL:*xx*. command.

VNP — System pool allocations are arbitrarily limited to 8192. bytes, but current request is for *n* bytes
This error can only be overcome by source modification. The request is made internally. You can regenerate the PE to allocate more pool or, perhaps, decrease the size of the Communications Executive partition. To allocate more pool, you must allocate less memory for those elements that take up pool space, such as CCBs.

VNP — The Communications Executive data structures are too large for VNP's internal buffer. The buffer (CET$$A *psect*) can be expanded to 2000 bytes maximum (from the current *n* bytes), but this configuration needs *m* bytes. This error is not recoverable without source modifications.
VNP's internal buffers cannot accommodate the Communications Executive Configuration File (CETAB). Reduce the size of the configuration.

VNP — UMR block allocation failure
There is not enough dynamic storage space to allocate a UMR block.

VNP — Unable to dynamically allocate network buffer pool
VNP could not allocate enough partition space to load the network pool.

VNP — Unit number too large
The unit number on the line identification is illegal.

VNP — Unspecified error during CONFIG file scan
VNP could not successfully parse all the definitions in the CETAB source file.

VNP — User identification field too long
The user identification field is limited to 16 characters.

Appendix A
Sample PEGEN Listings and Related Items

This appendix provides a sample PEGEN dialog listing for a PE (A.1) and the configuration file produced as a consequence of the PEGEN (A.2).

A.1 Sample Pegen Listing

```
>@[137,10]PEGEN
>;
>; ================================================================
>;       *  RSX-11 / 3271 Protocol Emulator Generation Procedure  *
>;                          Version V3.0
>; ================================================================
>;
>; Copyright (C) 1981, 1982 by
>; Digital Equipment Corporation, Maynard, Mass.
>;
>; Generation procedure started at 15:08:35 on 21-SEP-82
>;
>* Do you want to see the PEGEN notes/cautions? [Y/N]:Y
>;
>; Note the following:
>; -------------------
>;
>; * If  you have  not already performed  a SYSGEN for your target system
>;   you must do so before continuing.
>;
>;
>; * The  privileged  tasks  generated  during  PEGEN  use  configuration
>;   dependent  parameters  produced  by  SYSGEN. For RSX-11M/M-PLUS, the
>;   RSXMC.MAC file and  the  RSX11M.STB  file for  the target system are
>;   necessary. If they are not  already on the  output device  under the
>;   proper UIC,  PEGEN will move them from the user's  system device  to
>;   the output device.
>;
>; * If your distribution  medium is the output device, it is recommended
>;   that you backup this device before  continuing.  This is recommended
>;   because the device must be write-enabled during  the PEGEN sequence.
>;   If  your  disk  is  an  RK05, certain files  will be deleted to make
>;   sufficent space on the disk.
>;
>; * If  you cannot arrange to have the kit media and the target disk  on
>;   the  same drives, you should use the logical device  assignments  to
>;   ease  PE maintenance. Four devices are  required for the  build -- a
>;   kit device,  a target  device,  a listing  device  and a map device.
>;
```

```
>;   For example, the  following  MCR  commands create a  typical  set of
>;   logical names for the generation:
>;
>;     .>ASN DK:=K1:
>;     .>ASN DB:=TG:
>;     .>ASN DB:=LS:
>;     .>ASN DB:=MP:
>;
>;   This assigns K1: as the kit device; TG: as the target device; LS: as
>;   the list device; and MP:  as the  map device. When  asked for device
>;   names during PEGEN, use the assigned name for the respective device.
>;
>; * The proper system libraries for the target system must be on the LB:
>;   device under the UIC [1,1] while  you  are  performing  your  PEGEN.
>;   These libraries include:
>;
>;         EXELIB.OLB
>;         EXEMC.MLB
>;         RSXMAC.SML
>;         SYSLIB.OLB.
>;         NOANSLIB.OLB
>;
>;   Note  that NOANSLIB.OLB  is only  required if  your target system is
>;   RSX-11M-PLUS.
>;
>; * The  slow  Task  Builder (STK) is required by PEGEN.  If this is not
>;   present  on  your  system,  see  the  RSX-11M/M-PLUS  Task  Builder
>;   Reference Manual for information on installing STK.
>;
>; * If tasks required by PEGEN are not installed on your host system,
>;   and you are logged in as a privileged user, PEGEN will install
>;   them automatically. The following tasks are required on your
>;   system:
>;         PIP, LBR, MAC, and STK
>;         LOA and UNL (if device drivers require loading)
>;         MOU and DMO (if disks/tapes require mounting)
>;         UFD         (if required UICs do not already exist)
>;         FLX         (if target system files on magnetic tape)
>;
>; * For questions that  can be  answered YES or  NO, NO  is the default.
>;   For  questions  with  defaults, the default is produced by  pressing
>;   a carriage  return (RET) in response  to  the  question.  Additional
>;   explanatory  text  for  each question  is  available  by  hitting the
>;   "ESCAPE" key.
>;
>; <EOS> Do you want to -
>*       <CR>-continue, E-exit, P-Pause [S]:
>;
>* 1.   Target system type (M,M-PLUS)? [S]:
>;
>; The target system is the system for which you are doing this PEGEN.
>; The target system type determines how the network software is
>; generated and what features are available.
>;
>* 1.   Target system type (M,M-PLUS)? [S]: M
>* 2.   Distribution device (DDNN)? [S]:
>;
>; Specify  the  device  on  which  the  distribution kit is located (for
>; example DK2). Optionally, the  colon  may  be  supplied (for  example,
>; DK2:). The default device is "SY:".
>;
>* 2.   Distribution device (DDNN)? [S]: DR0
>;
>PIP DR0:/FR

DR0: has 27879. blocks free, 103801. blocks used out of 131680.
Largest contiguous space = 2560. blocks
5492. file headers are free, 2607. headers used out of 8099.
```

A-2 Sample PEGEN Listings and Related Items

```
>;
>; Copying and renaming system-dependent files
>;
>SET /UIC=[1,1]
>PIP DR0:[130,10]NETLIB.MLB/NV=DR0:[130,10]NETLIB11M.MLB
>PIP DR0:[130,10]NETLIB.MLB;*/PU/NM
>PIP DR0:[130,24]CEX.OLB/NV=DR0:[130,24]CEX11M.OLB
>PIP DR0:[130,24]CEX.OLB;*/PU/NM
>PIP DR0:[132,24]NTL.OLB/NV=DR0:[132,24]NTL11M.OLB
>PIP DR0:[132,24]NTL.OLB;*/PU/NM
>PIP DR0:[132,24]NTIKLC.OBJ/NV=DR0:[132,24]NTIKLC11M.OBJ
>LBR DR0:[132,24]NTI11M.OLB/RP/-EP=DR0:[132,24]NTIKLC
Module "NTIKLC" replaced

>PIP DR0:[132,24]NTIKLC.OBJ;*/PU/NM
>PIP DR0:[135,24]NCPMAP.OLB/NV=DR0:[135,24]NCPMAP11M.OLB
>PIP DR0:[135,24]NCPMAP.OLB;*/PU/NM
>PIP DR0:/FR

DR0: has 27877. blocks free, 103803. blocks used out of 131680.
Largest contiguous space = 2560. blocks
5492. file headers are free, 2607. headers used out of 8099.

>;
>; The following is assumed:
>;
>;      1.      Target system is RSX-11M.
>;      2.      The system is mapped.
>;
>;
>; ================================ ====================================
>;               *     PEGEN  PRELIMINARY     *
>; ================================ ====================================
>;
>; Target system characteristics -
>; ------------------------------
>*  1.  Processor type [S R:5-6]:
>;
>; Enter the target processor type e.g. 11/34
>;
>*  1.  Processor type [S R:5-6]: 11/44
>*  4.  More than 124K words of memory? [Y/N]:
>;
>; If your RSX11M system was generated with  this feature, answer YES
>; to this question.
>;
>*  4.  More than 124K words of memory? [Y/N]: N
>*  5.  Multi-user protection? [Y/N]:
>;
>; If your RSX-11M system was generated with this feature, answer YES
>; to this question.
>;
>*  5.  Multi-user protection? [Y/N]: Y
>*  7.  Powerfail recovery support? [Y/N]:
>;
>; If your RSX-11M system was generated with this feature, answer YES
>; to this question.
>;
>*  7.  Powerfail recovery support? [Y/N]: Y
>*  8.  Dynamic checkpoint allocation? [Y/N]:
>;
>; If your RSX 11M system was generated with this feature, answer YES
>; to this question.
>;
```

```
>*  8.  Dynamic checkpoint allocation? [Y/N]: /
>;
>; PEGEN procedure parameters -
>; -------------------------------
>*  1.  Target system device (DDNN)? [S]:
>;
>; This device will contain all  generated  and  configuration
>; dependent files. * NOTE * -- This must be a FILES-11 device.
>; The  default device is "SY:".
>;
>*  1.  Target system device (DDNN)? [S]: DR0
>*  2.  Is DR0: mounted? [Y/N]:
>;
>; If you have mounted DR0: or it has been mounted as a public
>; device, answer the question YES.
>;
>*  2.  Is DR0: mounted? [Y/N]: Y
>*  6.  Group UIC for output? [0 R:1-377]:
>;
>; This is the group code under which all generated files will
>; reside. This feature will allo  the user to do a number  of
>; PEGENs, with outputs separated by UIC.
>;
>*  6.  Group UIC for output? [0 R:1-377]: 370
>;
>;       The files RSX11M.STB and RSXMC.MAC are required.
>*  7.  Can the files be transfered by PIP? [Y/N]:
>;
>; The  system files RSX11M.STB and RSXMC.MAC are required. If
>; the input  device is any FILES-11 device then answer "Yes".
>;
>*  7.  Can the files be transfered by PIP? [Y/N]: /
>*  9.  What device are the system files on (DDNN)? [S]:
>;
>; This is the storage device (for example, DX1) for the system
>; files: RSXMC.MAC and RSX11M.STB. The default device is "SY:".
>;
>*  9.  What device are the system files on (DDNN)? [S]: DR0
>* 10.  Under which UIC is the file RSX11M.STB [X,Y]? (D:[1,54]) [S]:
>;
>; Specify where the file RSX11M.STB is located on DR0:
>;       (for example, "[1,54]").
>;
>* 10.  Under which UIC is the file RSX11M.STB [X,Y]? (D:[1,54]) [S]:
>* 11.  Under which UIC is the file RSXMC.MAC [X,Y]? (D:[11,10]) [S]:
>;
>; Where is RSXMC.MAC located on DR0: (for example, "[11,10]").
>;
>* 11.  Under which UIC is the file RSXMC.MAC [X,Y]? (D:[11,10]) [S]:
>* 12.  Listing device (DDNN)? [S]:
>;
>; The listing device should  either be a FILES-11 device or LP:.
>; All listing files will be generated on this device. By default
>; no listing files will be generated.
>;
>* 12.  Listing device (DDNN)? [S]: DR0
>* 13.  Map device (DDNN)? [S]:
>;
>; The map  device  should either  be a  FILES-11 device or LP:.
>; All map files will be generated on this device. By default no
>; map files will be generated.
>;
>* 13.  Map device (DDNN)? [S]: DR0:
>* 14.  Spool listings and/or maps? [Y/N]:
>;
>; If you want the listing and/or the maps spooled, answer "Yes".
>;
```

A-4 Sample PEGEN Listings and Related Items

```
>* 14.  Spool listings and/or maps? [Y/N]: N
>* 15.  Delete build files after use? [Y/N]:
>;
>; It is  recommended that  you have  at least  1000  free blocks
>; available on the  output  device, if you choose not to delete.
>; An additional 600  blocks  should  be available  if  maps  and
>; listings are desired.
>;
>* 15.  Delete build files after use? [Y/N]: N
>;
>; <EOS> Do you want to -
>*       <CR>-continue, R-repeat section, E-exit, P-Pause [S]:
>;
>; ==========================================================================
>;                    *     COMPONENT  SELECTION    *
>; ==========================================================================
>;
>; Component selection -
>; --------------------
>; Do you want -
>* 1.   The complete 3271 PE system? [Y/N]:
>;
>; Answer Yes to get a complete 3271 PE system. This includes the
>; Comm Exec, the processes: TUD,  AUX,  BSD, and  DDM's, and the
>; Utilities: NCP, NTL, NTINIT, VNP, CFE, NDA, DCE, LBK.
>;
>; NOTE: Trace is required for LBK. If Trace is not selected your
>;       LBK program will not function properly.
>;
>; Do you want -
>* 1.   The complete 3271 PE system? [Y/N]:
>* 2.   The Trace program? [Y/N]:
>;
>; The TRACE program provides a realtime trace of the activity on
>; a 3271 line and formats  the activity  for later  study. It is
>; useful for debugging  and to  see the actual  line activity as
>; seen  by  the  protocol  emulator. It  formats  all characters
>; collected by the traced  unit,  both received  from the master
>; and data and protocol messages  generated by  the  user or the
>; protocol  emulator. TRACE  MUST  be  selected  for LBK to work
>; properly.
>;
>; Do you want -
>* 2.   The Trace program? [Y/N]: Y
>* 13.  Should all tasks be built for the same partition? [Y/N]:
>;
>; If all of the network tasks are to be built to run in the same
>; partition,  you may  enter  the name of the partition here and
>; reduce the number of questions that will be asked during PEGEN.
>;
>* 13.  Should all tasks be built for the same partition? [Y/N]: Y
>* 14.  What is the partition name? [S RIO-6]:
>;
>; Enter the  name of the  partition to  be used  by  the Network
>; tasks. The default name (<RET>) is "GEN".
>;
>* 14.  What is the partition name? [S RIO-6]:
>;
>; <EOS> Do you want to -
>*       <CR>-continue, R-repeat section, E-exit, P-Pause [S]:
>;
>;
```

```
>;  ================================================================
>;                    *   CONFIGURATION  DATABASE   *
>;  ================================================================
>;
>;
>; Database parameters -
>; -------------------
>*  1.  Number of control buffers (CCBs)? [D R:4.-200.]:
>;
>;  Control buffers (communications control blocks or CCBs) are
>;  used by network processes to  communicate with and schedule
>;  each other.
>;
>*  1.  Number of control buffers (CCBs)? [D R:4.-200.]: 50
>*  2.  Number of large buffers (LDBs)? [D R:4.-200.]:
>;
>;  Large  buffers (LDBs)  are  used for the  transmission  and
>;  reception of data.
>;
>*  2.  Number of large buffers (LDBs)? [D R:4.-200.]: 50
>*  3.  Large buffer (LDB) size? [D R:192.-1024. D:280.]:
>;
>;  The 3271 PE  can use  multiple LDB's  to buffer  user data.
>;  However on a PDP-11/70 with a DV, the minimum LDB  size  is
>;  254. bytes. The LDB size will affect performance.  Choosing
>;  the LDB size involves a trade-off between  available memory
>;  and system performance. Since the  3271 PE  needs a certain
>;  number  of  buffers per  line, you  should  choose  a large
>;  enough LDB to hold typical messages, yet leave  enough pool
>;  to have approximately two LDB's per line.
>;
>*  3.  Large buffer (LDB) size? [D R:192.-1024. D:280.]:
>*  4.  Number of large buffers reserved for receives? [D R:1.-49. D:3.]:
>;
>;  Large buffers (LDBs)  are  used  for  both transmitting and
>;  receiving data segments. If the  pool size  falls below the
>;  threshold value, then only  receive  data  buffer  requests
>;  will be honored.
>;
>*  4.  Number of large buffers reserved for receives? [D R:1.-49. D:3.]: 20
>*  8.  Should the network be loaded top down in GEN? [Y/N]:
>;
>;  Answer YES to the following question if you want the Network
>;  software  to  be  allocated  from  the  top  of  the  GEN
>;  partition. This helps prevent partition fragmentation.
>;
>*  8.  Should the network be loaded top down in GEN? [Y/N]: Y
>;
>; Line parameters -
>; ---------------
>;
>; Device driver parameters -
>* 1.   Device Driver Process name [S R:0-3]:
>;
>;  The  Device Driver (DDM) Process names correspond to the two
>;  or three character device name. Enter a carriage return when
>;  there are no more names.
>;
>;  Legal DDM process names are:
>;
>;  Process Name      Type          MUX?       DMA?
>;  ------------      ----          ----       ----
>;      BU            Sync          No         No
>;      KBU           Sync          Yes        Yes
>;      DV            Sync          Yes        Yes
>;
>;
```

```
>; Device driver parameters -
>*  1.  Device Driver Process name [S R:0-3]: KBU
>*  2.  Number of KBU controllers? [D R:1.-16. D:1.]:
>;
>;  Enter  the  number  of KBU controllers that are installed  on
>;  the target system. Do  not include any controllers which were
>;  specified during target system  SYSGEN, since devices can not
>;  be shared between the PE and the Operating System.
>;
>*  2.  Number of KBU controllers? [D R:1.-16. D:1.]:
>*  4.  What is the KBU device priority [O R:4-6 D:5]:
>;
>;  Enter the hardware device priority level
>;  of all KBUs on the target system.
>;
>*  4.  What is the KBU device priority [O R:4-6 D:5]:
>;
>; KBU-0 parameters -
>*  5.  Number of lines? [D R:1.-6. D:1.]:
>;
>;  For  multiplexer  devices, respond to the following question
>;  with the number of lines for controller KBU-0.
>;
>*  5.  Number of lines? [D R:1.-6.] D:1.]:
>*  6.  Vector address? [O R:0-774 D:0]:
>;
>;  If  the  interrupt  vector  for KBU-0 is not known, answer
>;  the following question with a "0". The vector address may  be
>;  overridden with the NCP or VNP utility  when  the  line is
>;  loaded.  NOTE: Vector addresses must be entered in octal.
>;
>*  6.  Vector address? [O R:0-774 D:0]: 340
>*  7.  CSR address? [O R:160000-177777 D:177777]:
>;
>;  If  the  device  control  and  status  register  (CSR)  address
>;  for KBU-0  is  not  known  answer  the  following question with  a
>; "177777". The CSR address may  be  overridden  with  the  NCP
>;  or  VNP utility when the line is loaded. NOTE: The 16-bit
>;  virtual CSR address must be entered in octal.
>;
>*  7.  CSR address? [O R:160000-177777 D:177777]: 160060
>;  Line KBU-0-0
>*  9.  Unit CSR address? [O R:160000:177777 D:177777]:
>;
>;  For the KBU device, you should enter the CSR address of the
>;  DUP-11 which the KMC-11 is controlling.
>;
>*  9.  Unit CSR address? [O R:160000:177777 D:177777]: 160100
>* 10.  Number of stations? [D R:1.-32. D:1.]:
>;
>; Respond to the question with the number of device station
>; addresses that KBU-0-0 will be polled for, by the host system.
>;
>* 10.  Number of stations? [D R:1.-32. D:1.]:
>* 11.  Set line state to "ON" on loading the network? [Y/N]:
>;
>;  If KBU is set to "ON" the KBU device driver (DDM) and
>;  the  line's  data  base  will  be  automatically  loaded  into
>;  memory  at  network  initialization  time. The line state  will
>;  then  be  set "ON" and the network  software  (once  initialized)
>;  will  attempt  node  intialization  on  that  line.  If  it  is not
>;  enabled  the  user  may  set  the  state  of  the  line  to "ON" with
>;  either  the  NCP or  VNP  utility.  If the line is  multi-point, all
>;  tributaries on  the  line will  also be set to "ON".
>;
```

```
>* 11.  Set [illegible] state to "ON" on loading the network? [Y/N]: N
>* 12.  Control[illegible] address for KBU-0-0? [S R:4-5]:
>;
>; This  question  sets  the  controller  address  that  this  line
>; will  recognize  during  polling  by  the  host  system.  The
>; controller address should be specified with the symbol STAnumber,
>; where STAnumber is the symbol on the same row with the correct
>; address character in the following table:
>;
>; CU OR#        EBCDIC                          CU        CU
>; DEVICE        I/O      HEX      OCTAL        NUMBER    NUMBER
>; SYMBOL        CHAR.                          OCTAL     DECIMAL
>;
>; STA0          SP       40       100          0         0
>; STA1           A       C1       301          1         1
>; STA2           B       C2       302          2         2
>; STA3           C       C3       303          3         3
>; STA4           D       C4       304          4         4
>; STA5           E       C5       305          5         5
>; STA6           F       C6       306          6         6
>; STA7           G       C7       307          7         7
>; STA10          H       C8       310          10        8
>; STA11          I       C9       311          11        9
>; STA12         CENT     4A       112          12        10
>; STA13          .       4B       113          13        11
>; STA14          <       4C       114          14        12
>; STA15          (       4D       115          15        13
>; STA16          +       4E       116          16        14
>; STA17         BAR      4F       117          17        15
>; STA20          &       50       120          20        16
>; STA21          J       D1       321          21        17
>; STA22          K       D2       322          22        18
>; STA23          L       D3       323          23        19
>; STA24          M       D4       324          24        20
>; STA25          N       D5       325          25        21
>; STA26          O       D6       326          26        22
>; STA27          P       D7       327          27        23
>; STA30          Q       D8       330          30        24
>; STA31          R       D9       331          31        25
>; STA32          !       5A       132          32        26
>; STA33          $       5B       133          33        27
>; STA34          *       5C       134          34        28
>; STA35          )       5D       135          35        29
>; STA36          ;       5E       136          36        30
>; STA37          ~       5F       137          37        31
>;
>; The answer for station 0 would be STA0
>;
>* 12.  Controller address for KBU-0-0? [S R:4-5]: STA0
>* 13.  Station address for KBU-0-0 station 1, Unit TU1: [S R:4-5]:
>;
>; The  following  question  is  asked  for  each  station  on  the
>; controller. You will be asked for 1 unique  STAnumbers  for this
>; controller.  Please  answer  the  question as for the controller
>; address (e.g. STA0)
>;
>* 13.  Station address for KBU-0-0 station 1, Unit TU1: [S R:4-5]: STA1
>;
>; Device driver parameters -
>*  1.  Device Driver Process name [S R:0-3]:
>;
>; 3271 PE Line parameters
>; ----------------------
```

```
>* 1.   Number of NAk's received before abort? [D R:3.-30. D:7.]:
>;
>; Specify the number of  NAks  that can be  received  by  the 3271 PE
>; before  the  host  system  aborts  transmission  of  the  current
>; message  segment (at which  point you will be  informed of an error
>; condition).
>;
>* 1.   Number of NAk's received before abort? [D R:3.-30. D:7.]:
>* 2.   Number of ENQs to be sent on error? [D R3.-30. D:7.]:
>;
>; Specify  the number  of ENQs  to be sent by the 3271 PE to the host
>; system if a receive time out should occur or a receive error occurs
>; for a protocol message.
>;
>* 2.   Number of ENQs to be sent on error? [D R3.-30. D:7.]:
>* 3.   Number of SYNs on messages? [D R:3.-15. D:3.]:
>;
>; This is the number of SYNs to be sent before each data and  protocol
>; message. It may  be  necessary to  increase the number to compensate
>; for  bad  lines  or  to  delay  message  transmission. The latter is
>; desirable if, for example, the host system is slow in posting a read
>; for the 3271 PE's responses. The protocol requires a minimum of 3.
>;
>* 3.   Number of SYNs on messages? [D R3.-15. D:3.]:
>;
>; <EOS> Do you want to -
>*       <CR>-continue, R-repeat section, E-exit, P-Pause [S]:
>;
>; ================================================================
>;               *       COMPONENT  PARAMETERS   *
>; ================================================================
>;
>; Communications executive (CEX) -
>; ------------------------------
>*  2.  Communications Executive's partition name? [S R:0-6]:
>;
>; The Communications Executive provides:
>;
>;    - Interprocess communications and scheduling services
>;    - General  timer service for communications processes
>;    - Resource management of common buffer pools
>;
>;  Enter  the name of the partition where the Comm Exec (CEX) is  to  be
>;  loaded. The default partition name is "CEXPAR". This  partition  must
>;  be in the RSX Executive's address space.
>;
>*  2.  Communications Executive's partition name? [S R:0-6]:
>*  3.  CEXPAR partition base address? [O R:14000-117000]:
>;
>;  Enter  the  base  address  (in octal) of the CEXPAR  partition on the
>;  target  system.  The  Communications  Executive  must  reside  below
>;  120000 on  20K-Executive  systems  and  below 100000 on 16K-Executive
>;  systems.
>;
>*  3.  CEXPAR partition base address? [O R:14000-117000]: 112000
>*  4.  CEXPAR partition length? [O R:2700-6000]:
>;
>;  Enter  the  length  (in octal) of the CEXPAR partition on the target
>;  system.
>;
```

```
>*  4.  CEXPAR partition length? [D R:2700-6000]: 6000
>*  5.  Does the target system have a KG-11? [Y/N]:
>;
>;  The  KG-11  communications  arithmetic  option  computes  the  cyclic
>;  redundancy check characters which are used  for  error  detection  in
>;  serially  transmitted / received data. If the target system does  not
>;  have  the  KG-11  option  and  the communications interface option(s)
>;  installed on the  arget system do not perform crc  checking,  then  a
>;  software algorithm will be used.
>;
>*  5.  Does the target system have a KG-11? [Y/N]:N
>;
>; LLC processes -
>; -------------
>; TUD parameters -
>* 2. Maximum transmission block size? [D R:100.-1920. D:280.]:
>;
>; The  question  asks  for  the  maximum  transmit  block  size  for
>; the protocol emulator. This will set the maximum block size to the
>; value  given  except  on a UMR mapped system with a DV or KBU.  In
>; those systems, the maximum  transmit block size is dynamically set
>; to  the  receive  data  buffer  size  in  CETAB.   For  all  other
>; configurations  the  value  given  here  is the maximum before the
>; application gets an IE.RBG error on a write QIO.
>;
>* 2. Maximum transmission block size? [D R:100.-1920. D:280.]:
>;
>; Device counters Utility (DCE)
>; ----------------------------
>* 1. Do you wish the log file to be SY0:[370,54]TULOG.TXT? [Y/N]:
>;
>; Do  you  want  the  DCE  log file to be SY:[370,54]TULOG.TXT? If you
>; wish it to be  something else, answer NO to this  question. Then you
>; will be asked for:
>; 1) A device to replace SY0:,
>; 2) A UIC in the form X,Y (ie. 370,54) and
>; 3) A file name  in the form NAME.EXT (ie. TULOG.TXT)
>;
>* 1.   Do you wish the log file to be SY0:[370,54]TULOG.TXT? [Y/N]:
>;
>;
>; Screen Emulator (SCR)
>; --------------------
>* 2.   Number of read buffers ? [D R:3.-15. D:3.]:
>;
>; Enter  the number  of read  buffers to be used by the SCR utility. The
>; default  value  is  3.  The  number  of  buffers  is  dependent on the
>; application running but the default value is usually applicable unless
>; expect  the host system to send large blocks of  unsolicited data. For
>; example,  a value of 14 is more  suitable if you are running under VM.
>; Note  that  the  size  of SCR is  increased by about 2k bytes for each
>; additional read buffer.
>;
>* 2.   Number of read buffers ? [D R:3.-15. D:3.]:
>* 3.   First TU unit available for SCR [0 R:1-2 D:2]:
>;
>; Specify the range of TU units that can be used by the SCR utility.
>;
>; By default, the utility will select any free TU unit in the full
>; range available, (TU1 through TU2)
>;
>; If you wish the selection to be limited, specify the first and the
>; last units in a range. For example, if you wish to reserve some
>; TU units for specific applications, specify a range that does
>; not include these units.
>;
```

```
>* 3.   First TU unit available for SCR [0 R:1-2 D:1]:
>* 4.   Highest TU unit available for SCR [0 R:1-2 D:2]:
>;
>; Specify the range of TU units that can be used by the SCR
>; utility.
>;
>; By default the utility will select any free TU unit in the full
>; range available, (TU1 through TU2).
>;
>; If you wish the selection to be limited, specify the first and the
>; last units in a range. For example, if you wish to reserve some
>; TU units for specific applications, specify a range that does
>; not include these units.
>;
>* 3.   First TU unit available for SCR [0 R:1-2 D:1]:
>* 4.   Highest TU unit available for SCR [0 R:1-2 D:1]:
>;
>;
>; <EOS> Do you want to -
>*       <CR>-continue, R-repeat section, E-exit, P-Pause [S]:
>;
>; ========================================================================
>;                  *   BUILD 3271 PE COMPONENTS   *
>; ========================================================================
>;
>; All questions have been answered. PEGEN will  now  proceed  to  build
>; your system  the way  you  have chosen.   This  can take anywhere from
>; from 5 minutes to 3 hours,  depending on your selections and your host
>; system configuration.
>;
>; Take a break ...
>;
>; Build procedure started at 15:26:51 on 14-JUN-82.
>;
>SET /UIC=[370,24]
>MAC @DR0:CETABSM
>TKB @DR0:CEXBLD
>PIP DR0:[370,54]CEX.TSK/PU,CEX.STB
>TKB @DR0:STCRCBLD
>PIP DR0:[370,54]STCRC.TSK/PU,STCRC.STB
>TKB @DR0:DDHARBLD
>PIP DR0:[370,54]DDHAR.TSK/PU,DDHAR.STB
>TKB @DR0:AUXBLD
>PIP DR0:[370,54]AUX.TSK/PU,AUX.STB
>MAC DR0:TUDX,DR0:[370,34]TUDX/-SP=DR0:[130,10]NETLIB/ML,DR0:[370,10]PE,TUDX
>MAC DR0:TTB,DR0:[370,34]TTB/-SP=DR0:[130,10]NETLIB/ML,DR0:[370,10]PE,TTB
>TKB @DR0:TTBBLD
>PIP DR0:[370,54]TUD.DAT/NV=DR0:[131,24]TUD.DAT
>TKB @DR0:TUDBLD
>PIP DR0:[370,54]TUD.TSK/PU,TUD.STB
>PIP DR0:[370,10]/NV/FO=DR0:[131,10]BSCBUF.MAC
>MAC DR0:BSCBUF,DR0:[370,34]BSC/-SP=DR0:[130,10]NETLIB/ML,DR0:[370,10]PE,BSCBUF
>TKB @DR0:BSCBLD
>PIP DR0:[370,54]BSC BUF TSK/PU,BSCBUF.STB
>MAC DR0:BSDX,DR0:[370,34]BSDX/-SP=DR0:[130,10]NETLIB/ML,DR0:[370,10]PE,BSDX
>PIP DR0:[370,54]BSD.DAT/NV=DR0:[131,24]BSDX.DAT
>PIP DR0:[370,54]MODETB.TSK/NV/CO=DR0:[131,24]MODETBX.TSK
>TKB @DR0:BSDBLD
>PIP DR0:[370,54]BSD.TSK/PU,BSD.STB
>TKB @DR0:KBUBLD
>PIP DR0:[370,54]/NV=DR0:[131,24]KBU.DAT
>PIP DR0:[370,54]/NV=DR0:[131,24]KBULN.DAT
>PIP DR0:[370,54]KBU.TSK/PU,KBU.STB,*.DAT
>TKB @DR0:NTINITBLD
>PIP DR0:[370,54]NTINIT.TSK/PU
```

```
>TKB @DR0:NTLBLD
>PIP DR0:[370,54]NTL.TSK/PU
>TKB @DR0:[370,24]NCPBLD
>PIP DR0:[370,54]NCP.TSK/PU/NM
>TKB @DR0:CFEBLD
>PIP DR0:[370,54]CFE.TSK/PU
>TKB @DR0:[370,24]TRABLD
>PIP DR0:[370,54]TRA.TSK/PU
>MAC DR0:[370,24]DCEX,DR0:[370,34]D/-SP,=DR0:[130,10]NETLIB/ML,DR0:[370,10]PE,D
>TKB @DR0:[370,24]DCEBLD
>PIP DR0:[370,54]DCE.TSK/PU
>TKB @DR0:[370,24]LBKBLD
>PIP DR0:[370,54]LBK.TSK/PU
>TKB @DR0:NDABLD
>PIP DR0:[370,54]NDA.TSK/PU
>MAC DR0:3271EM,DR0:[370,34]3271EM/-SP=DR0:[370,10]SCRPRE,DR0:[131,10]3271EM
>TKB @DR0:[370,24]SCRBLD
>PIP DR0:[370,54]SCR.TSK/PU
>SET /UIC=[370,24]
>TKB @DR0:VNPBLD
>PIP DR0:[370,54]VNP.TSK/PU
>PIP DR0:[370,*]*.*/PU
>;
>; The network HELP file library will be moved to DR0: under [1,2].
>; When DR0: is brought to your target system, use the LBR utility to
>; extract the BUILD procedure module from the library, and start the
>; procedure to move the necessary files to the LB: device. For example:
>;       >SET /UIC=[1,2]
>;       >LBR LB:BUILD.CMD=xx:NETHLP.ULB/EX:BUILD
>;       >@LB:BUILD
>;
>SET /UIC=[1,2]
>PIP DR0:/NV=DR0:NETHLP.ULB
>SET /UIC=[370,24]
>;
>; Build procedure ended at 15:55:00.
>;
>; ================================================================
>;                    * Your PEGEN is now complete *
>; ================================================================
>;
>@<EOF>
>
```

A.2 Configuration File for 3271 PE

```
        .TITLE  CETAB
        .IDENT /V03.00/
;
; COPYRIGHT (C) 1978, 1979, 1980, 1982 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
        .MCALL  SYS$DF
        SYS$DF  <R$$11M,M$$MGE>
        BUF$DF  20.,28.,20.,280.,0.,0.,15.
        PDV$DF  <AUX,TUD>,<BSD>,<KBU>
        SLT$DF  KBU,BSD,TUD,LF.TIM,0,0
        LLC$DF  AUX,ZF.LLC!ZF.TIM!ZF.MTM!ZF.MFL,0,0
        LLC$DF  TUD,ZF.LLC!ZF.TIM!ZF.MFL,0,1.,,TU
        DLC$DF  BSD,ZF.DLC!ZF.COU,0,0,1.
        DDM$DF  KBU,ZF.DDM!ZF.MUX!ZF.LMC,5,4.,1.
        CNT$DF  0,340,160060,5,,NX
        UNT$DF  0,177471,2,160100,,0,0
        PAR$DF  ,GEN,51.,TOP,1,0
        END$DF
        EXPPDT  <AUX,TUD>,<BSD>,<KBU>
        EXPSLT  <BSD>,<KBU>
        EXPLLT  <AUX,TUD>
        EXPSTT
        EXPPUT  <AUX,TUD>,<BSD>,<KBU>
        .LIST
        .END
```

Appendix B
Partition Sizes

3271 PE PARTITION SIZES and MEMORY REQUIREMENTS

Minimum partition sizes

These are the minimum partition sizes for the 3271 PE processes and utilities given the specified options. All numbers are in decimal words.

	RSX-11M	RSX-11M-PLUS	
Processes			
AUX	330	na	
AUX w/powerfail	384	384	
BSD	2820+*n*	2820+*n*	Variable See Note 1
BSD w/Trace support	3128+*n*	3128+*n*	Variable See Note 1
BU	1056	1056	
DLX (M-PLUS only)	na	2080	
DV BISYNC only	1536	1536	
DV Shared BISYNC and DDCMP	1696	1696	
KBU	1312	1312	
TUD	1120+*n*	1160+*n*	Variable See Note 2
TUD w/UMR mapping	1144+*n*	1184+*n*	Variable See Note 2.

(Continued)

Table (Cont.)

	RSX-11M	RSX-11M-PLUS	
Utilities and Tasks			
CFE	7936	8256	
DCE	4800	5248	
LBK	7104	7552	
NCP	7936	7968	
NDA	8512	8992	
NTINIT	1792	1824	
NTINIT w/KBU support	2752	2848	
NTL	7680	8128	
TRA	8000	8608	
VNP (M only)	12512	na	
Communications Executive	864	na	See Note 3
Network Pool (POOL..) resident			
STRCRC	128	128	
STCRC w/KG-11	60	60	
DVMOD	512	512	See Note 4
DVMOD w/Trace	640	640	See Note 4
RSX Executive Pool resident			
CETAB	*n*	*n*	Variable See Note 3
TTB	16+*n*	16+*n*	Variable See Note 5

Notes:

1. BSD size, *n* = 3. words per line plus 7. words per TU device.
2. TUD size, *n* = 8. words plus 2. words per TU device.
3. CETAB may reside in CEXPAR on an RSX-11M system if there is enough space. CETAB size, *n* = 11. words for each system line plus 8. words for each process.
4. DVMOD tables used with BU and DV only.
5. TTB size, *n* = 28. words for each TU device.

Appendix C
Device Settings

Table C-1 indicates the jumper options for the DV11 synchronous multiplexer. The DV11 and the modem controller must have contiguous CSR and vector assignments. There are no settings for the DU11, DUP11, and DUV11.

Table C-1: DV11 Parameter Selection Switches

3271 PE Operating Requirements	Parameter	Name	Switch		Setting
			Pack	Number	
Internal Baud Rate Optional	1200 Baud	Select A	S2	3	ON
		Select B	S2	4	ON
	2400 Baud	Select A	S2	3	ON
		Select B	S2	4	OFF
	4800 Baud	Select A	S2	3	OFF
		Select B	S2	4	ON
	9600 Baud	Select A	S2	3	OFF
		Select B	S2	4	OFF
Full/Half Duplex	Full-Duplex	HD3	S2	5	ON
		HD2	S2	6	ON
		HD1	S2	7	ON
		HD0	S2	8	ON
	Half-Duplex	HD3	S2	5	OFF
		HD2	S2	6	OFF
		HD1	S2	7	OFF
		HD0	S2	8	OFF
		Note: The diagnostic and loopback tests only run with the full-duplex setting.			

Table C-1 (Cont.): DV11 Parameter Selection Switches

3271 PE Operating Requirements	Parameter	Name	Switch		Setting
			Pack	Number	
Parity	No Parity	PI	S1	1	OFF
		EPE	S1	2	OFF
Character Length (Excluding Parity)	8 Bits/Char	WLS1	S1	3	OFF
		WLS2	S1	4	OFF
SYNC Requirement	2 SYNC REQ.	1 SYNC 00	S1	5	ON
		1 SYNC 01	S1	6	ON
		1 SYNC 02	S1	7	ON
		1 SYNC 03	S1	8	ON
SYNC Character Codes 226 octal	Desired Code	Sync A	S4	1 8	(As required OFF=Logical one)
62 octal	Desired Code	Sync B	S3	1 8	(As required OFF=Logical one)

Appendix D
Configuration File Editor Utility (CFE)

Use the Configuration File Editor Utility (CFE) to change or to display the parameters of the configuration file, CETAB.MAC, that was created during PE generation. Refer to Sections 2.2.5 and 6.6.3 for more information on CFE. The commands presented here make up a CFE subset that can be used with the 3271 PE.

Most CFE commands consist of a command name, a component upon which the command acts, and parameters selected for that component. Any keyword can be abbreviated to the first three characters. See the Preface for notation conventions. Continue a command on another line by typing a hyphen at the point of the break, followed by a carriage return. Take care to enter the blanks specified for the command, even if one occurs just before the hyphen.

D.1 Using the HELP Command

Use the HELP command for assistance in selecting CFE commands and options. Executing this command will cause the summary HELP text to be displayed.

Format: HELP

Use the HELP command with a selected CFE command (and component) to display a portion of the HELP file for that command (for example, HELP DEFINE LINE).

Command prompting provides additional assistance when using CFE commands. At any point where you can type a space, type a carriage return instead. This causes a prompt requesting the required information to appear.

For example:

```
DEFINE
Define component: LINE
Line identification: ETH 1
Define line options: STATE
```

If you need more help, you can press the ESCAPE key, and a shorter form of the HELP file will be displayed.

D.2 Invoking CFE

Before invoking CFE, you must know the device and UIC under which the configuration file resides. Following PEGEN, CETAB.MAC resides under the UIC:

[*X*,*54*]

where *X* is the network UIC selected during PEGEN.

Invoke CFE by typing:

Method 1. MCR)CFE

CFE then returns the prompt:

```
ENTER FILENAME:
```

Enter the name of the configuration file that you want to change. If you omit any or all of the filename specifications, the default is SY: [current UIC] CETAB.MAC.

Enter the CFE command immediately following the prompt.

Method 2. MCR)CFE *filename*

where *filename* is the name of the configuration file that you want to change.

CFE displays the CFE prompt. Enter the CFE command immediately following the prompt.

Method 3. MCR)CFE *@command file* /TR

where *command file* is the name of a file containing CFE commands and [/TR] is an optional switch that allows you to trace the commands as they are executed.

The first entry in the command file must contain the name of the configuration file you want to change. The remainder of the file should contain one or more CFE commands. You can nest command files.

D.3 Exiting CFE

Use the EXIT command or a CTRL Z to terminate an editing session and update the configuration file.

Format: EXIT or CTRL Z

Use the EXIT PURGE command to terminate an editing session, update the configuration file, and delete the old file.

Format: EXIT PURGE

You can use the KILL command to abort an editing session, but execution of this command will not cause the configuration file to be updated, and will not delete the old file.

Format: KILL

D.4 CFE Dialog Example

CFE is invoked using Method 1, described above.

```
> CFE
```

Configuration file to change is CETAB.MAC

```
ENTER FILENAME: CETAB.MAC
```

Display characteristics of all known lines

```
CFE> LIS KNO LIN
```

The display (two known lines):

```
Known line characteristics as of dd-mon-yy hh:mm:ss

Line = BU-0
 Controller CSR = 160070, Vector = 320, Priority = 5
 Duplex = Half, Speed = Synchronous
 Cost = 0, State = On

Line = DV-0-0
 Controller CSR = 175000, Vector = 0, Priority = 5
 Duplex = Half, Speed = Synchronous
 Cost = 0, State = Cleared
```

Change parameters for DV-0-0: new vector address and set the line's operational state to OFF.

```
CFE> DEF LIN DV-0-0 VEC 340 STA OFF
```

Display the characteristics of DV-0-0 to verify that correct changes have been made.

```
CFE>LIS LIN DV-0-0

Line characteristics as of dd-mon-yy hh:mm:ss

Line = DV-0-0
 Controller CSR = 17500, Vector = 340, Priority = 5
 Duplex = Half, Speed = Synchronous
 Cost = 0, State = Off
```

Return to MCR

```
CFE>EXIT
>
```

D.5 CFE Command Descriptions

Only those CFE commands that are valid with PE systems are listed on the following pages, alphabetically according to the command name and the components on which they operate.

Use the DEFINE LINE command to change the parameters set during PEGEN for a specific line or all known lines.

Format: DEFINE $\left\{\begin{matrix}\text{LINE } \textit{line-id} \\ \text{KNOWN LINES}\end{matrix}\right\}$ *line-option(s)*

Example of command: CFE>DEFINE LINE BU-0 PRIORITY 5

or

CFE>DEF LIN BU-0 PRIO 5

where:

line-id is the identification of the line (Section 2.2.2.3).

KNOWN LINES indicates all lines known to the executor node (those defined at PEGEN).

line-option(s) consists of one or more of the following:

[CONTROLLER] CSR *csr-address*	set the address of the first CSR for the line controller.
PRIORITY *priority*	indicates the hardware priority of the controller. This argument is a number from 4 to 7, given to a hardware device.
STATE *line-state*	set the line's operational state. The possible states are as follows: CLEARED The line is not loaded. OFF The line is loaded and not being used.

	ON The line is available for normal use with the exception of temporary overrides for service functions.
UNIT CSR *csr-address*	indicates the address of the first CSR for a device being controlled by a KMC-11.
VECTOR *vector-address*	indicates the vector address of the controller. This argument must be the hardware vector address selected on the device.

Use the DEFINE PROCESS command to change the load parameters for a specific process or all known processes. See Chapter 2 for a detailed discussion of the PE processes.

Format: DEFINE { PROCESS *process-name* / KNOWN PROCESSES } *process-option(s)*

Example of command: CFE>DEFINE PROCESS BSD STATE ON

or

CFE>DEF PRO BSD STA ON

where:

KNOWN PROCESSES — set the parameters for all known processes (those defined at PEGEN).

PROCESS *process-name* — is one of the following processes:

BSD (BISYNC Protocol Process)

TUD (Terminal Unit Device Driver)

AUX (Auxiliary Processor)

DLX (Direct Access Interface)

xxx (a device mnemonic: BU, KBU, or DV)

process-option(s) — consists of one or more of the following:

MAXIMUM CONTROLLERS *count* — sets the maximum number of hardware controllers for which the process is to allocate data space. The number is decimal in the range 1 to 64.

MAXIMUM LINES *number* — set the maximum number of lines for which the process is to allocate data space. The number is decimal in the range 1 to 64.

STATE
process-state

sets the process's operational state. The possible states are as follows:

CLEARED

The process is not available for use and must be manually loaded.

ON

The process is available for normal use and is automatically loaded.

OFF

The process is loaded but not available for use.

Use the DEFINE SYSTEM command to change the Communications Executive system parameters set during PEGEN.

Format: DEFINE SYSTEM *system-option(s)*

Example of command: `CFE>DEFINE SYSTEM LOCATION TOPDOWN`

where:

system-option(s) consists of one or more of the following:

LARGE BUFFER SIZE *number*

sets the size (in bytes) of the large buffer. The number is a decimal number in the range 256 to 4096.

[LOCATION] *location-option*

controls the allocation of dynamic memory.

NOTE

That this option is only allowed if the network pool is dynamically allocated (specified during PEGEN).

where:

FIRSTFIT

specifies that processes will be loaded in the first available space that is big enough.

TOPDOWN

specifies that processes will be loaded at the top of the partition.

MAXIMUM CONTROL BUFFERS *number*

sets the number of control buffers available for Communications Executive system use. The number is decimal in the range 1 to 200.

MAXIMUM LARGE BUFFERS *number*	sets the maximum number of large buffers available for communications. The number is decimal in the range 1 to 200.
MINIMUM RECEIVE BUFFERS *number*	sets the minimum number of receive available for communications. The number is decimal in the range 1 to (maximum large buffers −1).
POOL BLOCK-AREA *block-count*	is the number of blocks needed for the network pool. This is a decimal number in the range 0 to 128.
POOL BYTE-AREA *byte-count*	is the number of bytes needed in the network pool. This is a decimal number in the range 0 to 8192.
POOL PARTITION *partition-name*	indicates the name of the partition into which the network pool is loaded.

Use the LIST LINE command to display the characteristics or a summary (the default) of the line parameters set during PEGEN.

Format: LIST $\left\{\begin{matrix}\text{KNOWN LINES}\\ \text{LINE } \textit{line-id}\end{matrix}\right\}$ [*display-type*]

Example of command: `CFE>LIST LINE DU-0-0 CHARACTERISTICS`

where:

KNOWN LINES displays information for all known lines.

LINE *line-id* displays information pertaining to a specific line.

display-type indicates the type of information to display. The *display-types* are CHARACTERISTICS and SUMMARY. If no *display-type* is specified, the default is SUMMARY.

Example of display:

```
Line characteristics as of dd-mon-yy hh:mm:ss

Line = BU-0
     Controller CSR = 160140, Vector = 340, Priority = 5
     Duplex = Full, Speed = Synchronous
     Cost = 0, State = Cleared

Line = BU-1
     Controller CSR = 160150, Vector = 350, Priority = 5
     Duplex = Full, Speed = Synchronous
     Cost = 0, State = Cleared
```

CFE
LIST PROCESS

Use the LIST PROCESS command to display the characteristics or a summary (the default) of the process parameters set during PEGEN.

Format: LIST $\left\{\begin{matrix}\text{KNOWN PROCESSES} \\ \text{PROCESS } \textit{process-name}\end{matrix}\right\}$ [*display-type*]

Example of command:
```
CFE>LIST PROCESS TUD
```

or

```
CFE>LIS PRO TUD
```

where:

KNOWN PROCESSES	displays information for all known processes.
PROCESS *process-name*	displays information pertaining to a specific process.
display-type	indicates the type of information to display. The *display-types* are CHARACTERISTICS and SUMMARY. If no *display-type* is specified, the default is SUMMARY.

Example of display:

```
Process characteristics as of dd-mon-yy hh:mm:ss

Process = BU
          State = Cleared
          Maximum lines = 0, Maximum controllers = 0
```

Use the LIST SYSTEM command to display the characteristics or a summary (the default) of the system parameters set during PEGEN.

Format: LIST SYSTEM [*display-type*]

Example of command:

```
CFE> LIST SYSTEM
```

or

```
CFE> LIS SYS
```

where:

display-type indicates the type of display. The *display-types* are CHARACTERISTICS and SUMMARY. If no *display-type* is specified, the default is SUMMARY.

Example of display:

```
System characteristics as of dd-mon-yy hh:mm:ss

     Pool name = POOL.., Partition name = GEN
     Block-area extension = 52, Byte-area
     extension = 64
     Location = Topdown
     Maximum control buffers = 20, Maximum small
     buffers = 0
     Maximum large buffers = 35, Large buffer size = 290
     Receive buffer minimum = 3
```

Appendix E
Network Control Program Utility (NCP)

The Network Control Program (NCP) is a utility program that accepts terminal commands to load, control, monitor and test the PE software. You will find a general introduction to the Network Control Program in Chapters 2 and 6. The commands presented here are an NCP subset that can be used for the 3271 PE.

Most NCP commands consist of a command name, a component upon which the command acts, and selected parameters for that component. You can abbreviate keywords to the first three letters. See the Preface for notation conventions. You can extend a command to another line by typing a hyphen just before the carriage return. Take care to enter the blanks specified for the command even if one occurs just before the hyphen.

E.1 Using the Help Command

Use the HELP command for assistance in selecting NCP commands and options. Executing this command causes the summary HELP text to be displayed.

Format: HELP

The use of the HELP command with a selected NCP command and components causes the portion of the HELP file for that command to be displayed (HELP SET LINE).

Command prompting provides additional assistance when using NCP commands. At any point where you can type a space, you can type a carriage return instead. This causes a prompt requesting the required information to appear.

For example:

```
SET
Set component: LINE
Line identification: BU-1
Set line options: STATE
```

If you need more assistance, you can type the escape key, and a shorter form of the HELP file will be displayed.

E.2 Invoking NCP

Use one of the following three methods to invoke NCP:

Method 1. `MCR>NCP`

NCP then returns the prompt:

```
NCP>
```

Enter the NCP command immediately following the prompt.

If the NCP command is executed successfully, you will receive the prompt on the next line. If, however, the command is unsuccessful, you will receive an error message on the next line indicating the reason for the error, followed by the prompt.

Example:

```
MCR>NCP
NCP>SET SYSTEM ALL
NCP>SET LINE BU-4 STATE ON
NCP  - Unrecognized Device or Unit
NCP>
```

Method 2. `MCR>NCP command`

In this case you return to MCR after the NCP command is executed.

Method 3. `MCR>NCP @command-file`

where:

`command-file` is the name of a file containing NCP commands. You can nest the commands.

E.3 Exiting NCP

Enter one of the following commands to exit NCP:

EXIT or CTRL/Z

When NCP terminates, it sets an exit status that can be inspected using the normal RSX exit status handling techniques. (See the RSX documentation.) NCP sets one of four exit statuses to indicate the worst case of the applicable commands executed. These statuses, in order from best to worst case, and their meanings are:

Status	Numeric Value	Meaning
Success	1	Command executed as requested.
Warning	0	SHOW for a single component; component is not in ON or REACHABLE state.
Error	2	Command accepted by NCP but failed in execution, either completely or partially.
Severe error	4	Command not accepted by NCP, for example due to command syntax error or command read error.

The following commands do not affect the exit status:

EXIT or CTRL/Z
HELP

E.4 NCP Dialog Examples

The following NCP command sequences assume that there are two BUs and two KBU lines, and that line zero for each device was set to ON during PEGEN.

1. In order to bring up the PE system with lines BU-0 and KBU-0-0 ON:

```
MCR>NCP
NCP>SET SYSTEM ALL
NCP>^Z
```

2. In order to bring up the PE system with all four lines ON:

```
MCR>NCP
NCP>SET SYSTEM ALL
NCP>SET LINE BU-1 STATE ON
NCP>SET LINE KBU-0-1 STATE ON
NCP>^Z
MCR>
```

3. In order to turn off and unload line BU-0 and to load and turn on line BU-1:

```
MCR>NCP
NCP>CLEAR LINE BU-0 ALL
NCP>SET LINE BU-1 STATE ON
NCP>^Z
MCR>
```

4. In order to bring up the PE system with both BU lines ON but with neither KBU line ON:

```
MCR>NCP
NCP>SET SYSTEM ALL
NCP>CLEAR LINE KBU-0-0 ALL
NCP>SET LINE BU-1 STATE ON
NCP>^Z
MCR>
```

E.5 NCP Command Descriptions

Only those NCP commands that are valid with PE systems are listed below, alphabetically according to the command name and components on which they operate.

Use the CLEAR LINE ALL command to unload a line.

Format: CLEAR LINE *line-id* ALL

Example of command: NCP> CLEAR LINE BU-2 ALL

where:

LINE *line-id*	is the identification of the line to be unloaded. Section 2.2.2.3 specifies line identification format.
ALL	unloads all parameters associated with the line.

NCP

CLEAR PROCESS ALL

Use the CLEAR PROCESS ALL command to unload a process.

Note that you can unload a process only when it is not being used by the system.

Format: CLEAR PROCESS *process-name* ALL

Example of command:

```
NCP> CLEAR PROCESS TUD ALL
```

or

```
NCP> CLE PRO TUD ALL
```

where:

process name	is one of:
	AUX
	BSD (3271 Protocol Emulator)
	DLX (Direct Line controller -- RSX-11M-PLUS only)
	TUD (Terminal Unit Driver)
	xxx — (a device mnemonic: either BU, KBU or DV)
ALL	clears all parameters associated with the process.

Use the CLEAR SYSTEM ALL command to unload the PE Communications Executive and release those resources being used by the PE system.

Format: CLEAR SYSTEM [ALL]

Example of command: NCP> CLEAR SYSTEM ALL

NCP
SET KNOWN LINES

Use the SET KNOWN LINES command with ALL or the 'loading-options' (see below) to load the driver process and the appropriate data bases for all lines in the permanent data base. Use the SET KNOWN LINES command with the 'loaded options' to change parameters and state associated with loaded lines.

Format: SET KNOWN LINES $\left\{\begin{matrix}\text{ALL} \\ \textit{loading-option(s)}\end{matrix}\right\}$

Example of command: NCP>SET KNOWN LINES STATE OFF

where:

ALL — loads the lines with the defaults specified in the permanent data base for each of the line options.

loading-option(s) — consists of one or more of the options described below. To use any option except STATE, the line must be cleared.

[LOCATION] *location-option* — controls dynamic memory allocation. The *location-options* are:

FIRSTFIT

allocates the driver process to the first available space in the partition that is large enough.

TOPDOWN

loads the driver process at the top of the partition.

STATE *line-state* — sets the operational state of the lines to one of OFF, ON, or SERVICE (see SET LINE command).

Use the SET LINE command with ALL or the 'loading options' (see below) to load the driver process and the appropriate data bases for a line. With NCP, a line remains loaded until you either unload it with the CLEAR LINE ALL or CLEAR SYSTEM command, or reboot the system. Use the SET LINE command with the 'loaded options' to change parameters and state associated with loaded lines.

Format: SET LINE *line-id* { ALL / *loading-option(s)* / *loaded-option(s)* }

Example of command: `NCP> SET LINE DV-0-1 STATE ON`

where:

line-id is the identification of the line (Section 2.2.2.3).

ALL loads the line with the defaults specified in the permanent database for each of the line options.

loading-option(s) consists of one or more of the options described below. To use any option except STATE, the line must be cleared.

[CONTROLLER] CSR *csr-address*

sets the address of the first CSR for a controller device.

[LOCATION] *location-option*

controls the allocation of dynamic memory. The *location-*options are:

FIRSTFIT

allocates the driver process to the first available space that is large enough.

TOPDOWN

loads the driver process at the top of the partition.

	PRIORITY *hardware-priority*	sets the hardware priority given to the controller. *Hardware-priority* is a number (from 4-7) given a hardware device.
	STATE *line-state*	sets the operational state of a line. The states are:
		OFF
		the line is not available for any operation.
		ON
		the line is loaded and is in its normal operational state.
		SERVICE
		the line is loaded and is reserved for line service operations such as line loopback testing.
	UNIT CSR *csr-address*	sets the address of the first CSR for a device being controlled by a KMC-11 (KBU).
		NOTE That with the KBU, a unit CSR may be specified for any unit on the controller.
	VECTOR *hardware vector*	sets the vector address of the controller. This argument must be the hardware vector address selected on the device.
loaded-option	consists of the option below.	
	STATE *line-state*	sets the operational state of the line to one of OFF, ON, or SERVICE (see above).

Use the SET PROCESS command to load a process. You can use this command to override default parameters specified during PEGEN. With NCP, a process remains in memory until you clear the system or the process (see CLEAR SYSTEM and CLEAR PROCESS commands), or reboot the system.

Format: SET PROCESS *process-name* { ALL / *process-option(s)* }

Example of command:

```
NCP>SET PROCESS BSD LOCATION TOPDOWN
```

or

```
NCP>SET PRO BSD TOP
```

where:

process-name loads a process. The *process-name* options are:

AUX

TUD

BSD

DLX (RSX-11M-PLUS only)

xxx — (A device mnemonic: either BU, KBU or DV)

ALL loads the process with the defaults specified in the permanent database for each of the *process-options*.

process-option(s) consists of one or more of the following:

[LOCATION] *location-option* controls the allocation of dynamic memory. If omitted, FIRSTFIT is used. The *location-options* are:

FIRSTFIT

allocates the first available space that is large enough.

	TOPDOWN loads the driver process at the top of the partition.
MAXIMUM CONTROLLERS *count*	sets the maximum number of controllers the process can allocate space for. The count is a decimal number in the range 1 to 64.
MAXIMUM LINES *count*	sets the maximum number of lines the process can handle. The maximum is a decimal number in the range of 1 to 64. If omitted, the number specified in the Configuration File is used.
PARTITION *partition-name*	sets the name of the partition to override the partition name from the permanent data base.

Use the SET SYSTEM ALL command to load the RSX 3271 PE Communications Executive, and all network process and line defaults that were set in the permanent database. See Chapter 5.

Format: SET SYSTEM [ALL]

Example of command: NCP>SET SYSTEM ALL

NCP
SHOW LINE

Use the SHOW LINE command to display the status, characteristics counts and a summary (the default) for a specific line, only the active lines or all known lines on the local node. You can direct the output of the SHOW LINE command to the calling terminal or to any output file or device at the local node.

Format: SHOW { LINE *line-id* / ACTIVE LINES / KNOWN LINES } [*display-type*] [TO *output-file*]

Example of command: `NCP> SHOW KNOWN LINES STATUS`

or

`NCP> SHOW KNO LIN STA`

where:

LINE *line-id* displays information for a specified line.

ACTIVE LINES displays information for those lines in the ON or SERVICE state.

KNOWN LINES displays information on all known lines.

display-type is one of CHARACTERISTICS, COUNTERS, STATUS, or SUMMARY.

TO *output-file* directs the output to the specified file or device at the local node. The data is transferred in ASCII, in the same format as it is displayed at the terminal. If omitted, the information is printed at the calling terminal. If the specified output file already exists, the information appended to it.

Displaying Line Characteristics:

The format for displaying line characteristics is:

LINE CHARACTERISTICS AS OF *dd-mon-yy hh:mm*
LINE = *line-id*

COST = *cost*, TYPE = one of POINT (*point-to-point*), CONTROL (*multipoint control station*) or TRIBUTARY (*multipoint tributary*)
[TRIBUTARY = *tributary address*,] OWNER = *owner-id*
CONTROLLER CSR = *csr* [,UNIT CSR = *csr*]
VECTOR = *controller vector address*

PRIORITY = *priority*
[MULTIPOINT ACTIVE = *active ratio number*, MULTIPOINT DEAD = *dead ratio number*]

The information displayed reflects the actual values for lines that have been set. Lines that have not been set have no values.

The owner of a PE line is TUD, indicating the line is intended for use by the PE.

Example:

```
Line characteristics as of dd-mon-yy hh:mm
Line = BU-0

Cost = 1, Type = Point
Owner = TUD
Controller CSR = 160140
Vector = 330
Priority = 5
```

Displaying the Line Counters:

The format for displaying line counts is:

LINE COUNTERS AS OF *dd-mon-yy hh:mm*
LINE = *line-id*

count	counter-description
.	.
.	.
.	.

Example:

```
LINE COUNTERS AS OF dd-mon-yy hh:mm
LINE BU-0

  10   Seconds since last zeroed
 963   Data blocks received
 412   Data blocks sent
   0   NAKs transmitted
   0   NAKs received
   0   Acknowledgement timeouts
   0   WACKs transmitted
   0   NAK thresholds exceeded
   0   ENQ thresholds exceeded
   0   ACKs received out of sequence
   0   3271 command rejects
   0   Transmissions aborted by the host
   0   Receptions aborted by the host
   0   Transmit clock losses
   0   Receive clock losses
   0   Hardware errors
```

Table 6-2 of this manual describes the above counters.

Displaying the Status (or Summary) of Lines:

The format for displaying the status of lines is:

LINE STATUS AS OF *dd-mon-yy hh:mm*

LINE	STATE	LOOP BACK NAME	ADJACENT NODE	BLOCK SIZE

```
line-id state [substate]

   .              .

   .              .

   .              .
```

where:

line-id	*is the identification of the line.*
STATE	is the current state of the line.
LOOPBACK NAME	Not applicable.
ADJACENT NODE	Not applicable.
BLOCK SIZE	Not applicable.

A line that has not been set will have a state of CLEARED.

In PE systems, the ADJACENT NODE and LOOPBACK NAME fields will be blank. The summary output format for lines will be the same as the status output format except that block size is not included.

Example:

```
KNOWN LINES STATUS AS OF dd-mon-yy hh:mm

LINE     STATE

BU-0     ON-
DV-0-0   CLEARED
```

Use the SHOW PROCESS command to display information about a specific process, the active processes or all known processes. You can direct the output of the SHOW PROCESS command to the calling terminal or to any output file or device at the local node.

Format:

SHOW { PROCESS *process-name* / KNOWN PROCESSES / ACTIVE PROCESSES } { (STATUS) / [SUMMARY] } [TO *output-file*]

Example of command: NCP> SHOW PROCESS TUD

where:

STATUS or SUMMARY	displays the state of a process.
KNOWN PROCESSES	displays information about all processes.
ACTIVE PROCESSES	displays information about all active processes.
process-name	is one of the following: AUX TUD BSD DLX (RSX-11M-PLUS only) *xxx* — (a device mnemonic: BU, KBU or DV)
TO *output-file*	directs the output to the specified file or device at the local node. The data is transferred in ASCII in the same format as it is displayed at the terminal. If omitted, the information is printed at the calling terminal. If the specified output file already exists, the information will be appended.

Displaying the Status or Summary of Processes:

The format for displaying the status or summary of processes is:

KNOWN PROCESSES STATUS (or SUMMARY) AS OF
dd-mon-yy hh:mm

PROCESS	STATE
process-name	*state*
.	.
.	.

Example:

```
PROCESS STATUS AS OF dd-mon-yy hh:mm

    PROCESS STATE

    AUX     ON
    TUD     ON
    BSD     ON
    KBU     CLEARED
```

Use the SHOW SYSTEM command to display the system information. You can direct the output of the SHOW SYSTEM command to the calling terminal or to any output file or device at the local node. Table E-1 lists the standard text for system counters.

Format: SHOW SYSTEM [*display-type*] [TO *output-file*]

Example of command: NCP>SHOW SYSTEM COUNTERS

where:

display-type is one of COUNTERS, STATUS, CHARACTERISTICS, or SUMMARY.

TO *output-file* directs the output to the specified file or device at the local node. The data is transferred in ASCII in the same format as it is displayed at the terminal. If omitted, the information is printed at the calling terminal. If the specified output file already exists, the information is appended to it.

Example of the Display for the Contents of the System Counters:

```
SYSTEM COUNTERS AS OF dd-mon-yy hh:mm

 12    SECONDS SINCE LAST ZEROED
  0    CONTROL BUFFER ALLOCATION FAILED
  0    SMALL BUFFER ALLOCATION FAILED
  0    LARGE BUFFER ALLOCATION FAILED
  1    RECEIVE BUFFER ALLOCATION FAILED
```

Example of the Display for System Characteristics:

```
SYSTEM CHARACTERISTICS AS OF dd-mon-yy hh:mm

MAXIMUM CONTROL BUFFERS = 200, MAXIMUM SMALL BUFFERS = 0
MAXIMUM LARGE BUFFERS = 200, LARGE BUFFER SIZE = 5000
MAXIMUM RECEIVE BUFFERS = 200
```

Example of the Display for the System Status:

```
SYSTEM STATUS AS OF dd-mon-yy hh:mm

ACTIVE CONTROL BUFFERS = 41, ACTIVE SMALL BUFFERS = 0
ACTIVE LARGE BUFFERS = 20,
```

Example of the Display for a System Summary:

```
SYSTEM SUMMARY AS OF dd-mon-yy hh:mm

ACTIVE CONTROL BUFFERS = 20, ACTIVE SMALL BUFFERS = 0
ACTIVE LARGE BUFFERS = 12 MAXIMUM CONTROL BUFFERS = 150,
MAXIMUM SMALL BUFFERS = 0
MAXIMUM LARGE BUFFERS = 120
```

Table E-1 lists the RSX Communications Executive system counters.

Table E-1: System Counters

Standard Text	Description
Seconds since last zeroed	Records up to 65,534 seconds since its associated counters were zeroed. It is incremented by 2 every 2 seconds and is thus always an even number. Zeroed when its associated counters are zeroed, thus providing a time frame for them.
Control buffer allocation failed	Records up to 65,534 times that a control buffer was not available. Caused by insufficient control buffers to service the work load on the Communications Executive and its processes.
Small buffer allocation failed	Records up to 65,534 times that a small buffer was not available. Caused by insufficent small buffers to service the work load on the Communications Executive and its processes.
Large buffer allocation failed	Records up to 65,534 times that a large buffer was not available. Caused by insufficient large buffers available to service the work load on the Communications Executive and its processes. Can be a symptom of the minimum receive buffer level reserving too many large buffers.
Receive buffer allocation failed	Records up to 65,534 times that a receive buffer was not available. Caused by insufficient receive buffers reserved from the large buffers to service the work load on the Communications Executive and its processes.

NOTE

Any of the above system allocation failures can be resolved by using CFE to increase the supply of the resource, and then reloading the PE system.

Use the ZERO LINE command to zero all line counters associated with one line. Use the ZERO KNOWN LINES command to zero line counters for all known lines.

Format: ZERO $\left\{\begin{matrix}\text{KNOWN LINES} \\ \text{LINE } \textit{line-id}\end{matrix}\right\}$ [COUNTERS]

Example of command: NCP>ZERO LINE BU-2 COUNTERS

where:

line-id	is the identification of the line.
COUNTERS	is an optional keyword.

NCP
ZERO SYSTEM

Use the ZERO SYSTEM command to zero the system counters.

Format: ZERO SYSTEM [COUNTERS]

Example of command: NCP>ZERO SYSTEM

or

NCP>ZER SYS

where:

COUNTERS is an optional keyword.

Appendix F
Virtual Network Program Utility (VNP)

The Virtual Network Program (VNP) is a utility program that accepts terminal commands (from an RSX-11M or RSX-11m-PLUS operating system) to change the PE software configuration on an RSX-11M disk image file. Chapters 2 and 6 introduce VNP. The commands given here are a subset of VNP commands that may be used with PE systems.

Most VNP commands consist of a command name, a component upon which the command acts, and selected parameters for that component. Any keyword can be abbreviated to the first three letters. See the Preface for notation conventions. You can extend a command to another line by typing a hyphen just before the carriage return. Take care to enter the blanks specified for the command, even if one occurs just before the hyphen.

F.1 Using the Help Command

Use the HELP command for assistance in selecting VNP commands and options. Execution of this command causes the summary HELP text to be displayed.

Format: HELP

The use of the HELP command with a selected VNP command (and component) causes a portion of the HELP file for the command to be displayed (HELP SET LINE).

Command prompting provides additional assistance when using VNP commands. At any point where you can type a space, you can type a carriage return instead. This causes a prompt requesting the required information to appear.

For example:

```
SET
Set component: LINE
Line identification: BU-1
Set line options: STATE
```

If you need more help, you can press the escape key, and a shorter form of the HELP file will be displayed.

F.2 Invoking VNP

When you invoke VNP from a RSX-11M/M-PLUS system you can alter an 11M disk image file. You cannot use VNP on a RSX-11M-PLUS disk image.

Use one of the following methods to invoke VNP:

Method 1. `MCR>VNP`

VNP returns the prompt:

```
ENTER FILENAME> filename
```

Enter the name of the operating system file you want to change. The device and UIC specifications are optional. If you omit the filename, the default operating system filename and UIC is SY: [*current* UIC] RSX11M.SYS.

VNP then displays the prompt:

```
VNP>
```

Enter the VNP command immediately following the prompt.

NOTE

In order to execute any VNP commands, it is required that both the operating system and the PE system tasks reside under the same device and UIC.

Method 2. `MCR>VNP filename`

where `filename` is the name of the operating system disk image file you want changed.

VNP displays the VNP prompt. Enter the VNP command immediately following the prompt.

Method 3. MCR>VNP @command file [/TR]

where command file is the name of a file containing VNP commands and [/TR] is an optional switch that allows you to trace the commands as they are executed.

The first entry in the command file must contain the name of the disk image file you want to change. The remainder of the file should contain one or more VNP commands. You can nest command files.

F.3 Exiting VNP

Enter one of the following commands to exit VNP:

EXIT or CTRL/Z

F.4 VNP Command Descriptions

The VNP commands are listed on the following pages alphabetically according to the command name and components on which they operate.

VNP
CLEAR LINE ALL

Use the CLEAR LINE ALL command to unload a network line.

Format: CLEAR LINE *line-id* ALL

Example of command: VNP>CLEAR LINE BU-1 ALL

where:

line-id is the identification of the line to be unloaded.

Use the CLEAR PROCESS ALL command to unload a PE process.

Format: CLEAR PROCESS *process-name* [ALL]

Example of command: VNP> CLE PRO TUD ALL

where:

process name is one of the following:

AUX

BSD

TUD

DLX (RSX-11M-PLUS only)

xxx - (a device mnemonic: either BU, KBU, or DV)

VNP
CLEAR SYSTEM ALL

Use the CLEAR SYSTEM ALL command to unload the RSX 3271 PE Communications Executive.

Format: CLEAR SYSTEM [ALL]

Example of command: VNP>CLE SYS ALL

Use the SET KNOWN LINES command with ALL or the 'loading-options' (see below) to load the driver process and the appropriate data bases for all lines in the permanent data base. The driver process will be loaded when the system is rebooted. Use the SET KNOWN LINES command with the 'loaded options' to change parameters and states in the system image file that are associated with loaded lines.

Format: SET KNOWN LINES $\left\{\begin{matrix}\text{ALL} \\ \textit{loading-option(s)}\end{matrix}\right\}$

Example of command: VNP> SET KNOWN LINES STATE OFF

where:

ALL — loads the lines with the defaults specified in the permanent data base for each of the line options.

loading-option(s) — consists of one or more of the options described below. To use any option except STATE, the line must be cleared.

[LOCATION] *location-option* — controls dynamic memory allocation of the driver process. The options are:

FIRSTFIT

allocates the first available space in the partition that is large enough.

TOPDOWN

loads the driver process at the top of the partition.

STATE *line-state* — sets the operational state of the lines to one of OFF, ON, or SERVICE (see SET LINE command).

VNP
SET LINE

Use the SET LINE ALL command or the 'loading option(s)' (see below) to load the driver process and the appropriate data bases for a line. The driver process is loaded when the system is rebooted. Use the 'loaded options' with the SET LINE command to change parameters and states in the system image file that are associated with loaded lines.

Format: SET LINE *line-id* { ALL / *loaded-option(s)* / loaded-option(s) }

Example of command: `VNP> SET LINE BU-1 STATE ON`

where:

line-id — is the identification of the line. Section 2.2.2.3 specifies the format for identifying lines.

ALL — loads the line with the defaults specified in the permanent database for each of the line options

loading-option(s) — consists of one or more of the options described below. To use any of these options except STATE, the line must be cleared.

|CONTROLLER| CSR *csr-address* — sets the address of the first CSR controller device.

|LOCATION| *location-option* — controls the allocation of dynamic memory. The location options are:

FIRSTFIT

allocates the first available space that is large enough.

TOPDOWN

loads the driver process at the top of the partition.

NOTE

If during PEGEN, you specified top-down loading, you must ensure that your disk image file is at least as large as the real memory size. Disk Image Size = Memory Size in K Words × 4 + 2. If this constraint is satisfied and the real memory size is equal to or less than 124K, top-down loading is supported.

PRIORITY *hardware-priority*

indicates the hardware priority given to the controller. *Hardware-priority* is a number (from 4-7) given a hardware device.

STATE *line-state*

sets the state of the line. This state is one of the following:

OFF

the line is not available for any operation.

ON

the line is in its normal operational state.

SERVICE

the line is reserved for line service operations such as loopback testing.

UNIT CSR *csr-address*

indicates the address of the first CSR for a device that is controlled by a KMC-11 (KBU).

NOTE

With the KBU, a unit CSR may be specified for any unit on the controller.

	VECTOR *hardware-vector*	indicates the vector address of the controller. This argument must be the hardware vector address selected on the device.
loaded-option(s)	consists of the option described below.	
	STATE *line-state*	see under loading-option(s).

Use the SET PROCESS command to load a process.

This command can be used to override default parameters (specified during PEGEN). If set by VNP, a process will be in memory when the system is rebooted.

Format: SET PROCESS *process-name* {ALL / *process-option(s)*}

Example of command: VNP> SET PROCESS TUD ALL

where:

process-name — the process-name options are:

AUX

BSD

DLX (RSX-11M-PLUS only)

TUD

xxx — (a device mnemonic: BU, KBU, or DV)

ALL — loads the process with the defaults specified in the permanent data base for each of the *process-options*.

process-option(s) — consists of one or more of the following:

[LOCATION] *location-option* — controls the allocation of dynamic memory. If omitted, FIRSTFIT is used. The location options are:

FIRSTFIT

allocates the first available space that is large enough.

TOPDOWN

loads the driver process at the top of the partition.

MAXIMUM CONTROLLERS *count*	sets the maximum number of hardware controllers for which the process is to allocate data space. The number is a decimal in the range 1 to 64.
MAXIMUM LINES *count*	sets the maximum number of lines the process can handle. The maximum is a decimal number in the range of 1 to 64. If omitted, the number specified in the Configuration File is used.
PARTITION *partition-name*	sets the name of the partition to override the partition name from the permanent data base.

Use the SET SYSTEM ALL command to load the RSX 3271 PE Communications Executive and all the PE processes and lines that were set in the permanent data base.

Format: SET SYSTEM [ALL] [ECHO]

Example of command: `VNP>SET SYSTEM ALL`

where:

ALL	is an optional keyword.
ECHO	displays the names of the processes and lines being loaded on the terminal as they are loaded.

VNP
SHOW LINE

Use the SHOW LINE command to display information about a specific line, or all known lines on the local node. (Refer to the NCP SHOW LINE command in Chapter 6 for the display formats.)

Format: SHOW $\left\{\begin{matrix}\text{LINE } \textit{line-id} \\ \text{KNOWN LINES}\end{matrix}\right\}$ [*display-type*]

Example of command: `VNP> SHOW LINE BU-0 SUMMARY`

where:

display-type	is one of CHARACTERISTICS or SUMMARY.
LINE *line-id*	displays information about a specified line.
KNOWN LINES	displays information about all known lines.

Use the SHOW PROCESS command to display information about a specific process or all known processes. (Refer to the NCP SHOW PROCESS command in Chapter 6 for the display formats.)

Format: SHOW { PROCESS *process-name* / KNOWN PROCESSES } *display-type*

Example of command: VNP> SHOW PROCESS BU-0 STATUS

where:

display-type	consists of one of CHARACTERISTICS or SUMMARY.
KNOWN PROCESSES	displays information about all processes.
process-name	is one of the following: AUX BSD TUD DLX (RSX-11M-PLUS only) *xxx* — (a device mnemonic: BU, KBU, or DV)

VNP

SHOW SYSTEM

Use the SHOW SYSTEM command to display the system information. (Refer to the NCP SHOW SYSTEM command in Chapter 6 for the display formats.)

Format: SHOW SYSTEM *display-type*

Example of command: VNP> SHOW SYSTEM

where:

display-type	is one of CHARACTERISTICS or SUMMARY.

Index

A

ACK (Positive Acknowledgement)
  defined, 7-8
  functions, 7-9
ASCII/EBCDIC translation, 1-5, 2-8
AUX (Auxiliary process), 6-6
  general description, 2-8
  parameters (PEGEN), 4-29
  partition size, B-1

B

BISYNC
  error handling, 8-2
BISYNC Messages, 7-10
BSD (3271 Protocol Emulation process), 6-6, 6-27 to 6-29
  general description, 2-8 to 2-9
  parameters (PEGEN), 4-30
  partition size, B-1
BU device driver, 6-6
  defined, 2-9
  LDBs for, 3-9, 3-10
  parameters (PEGEN), 4-31
  partition size, B-1
Buffers, 3-8 to 3-12
  modifying parameters (CFE), 6-27
  monitoring, 6-16
  pool, 3-8, 6-27

C

CCBs (Communication Control Blocks), 2-7, 3-2, 3-7, 3-8, 3-9
  monitoring, 6-16
  PEGEN requirements, 3-11 to 3-12
CETAB
  partition size, B-2
CETAB.MAC configuration file, 2-10, 2-14
  modifying with CFE, 6-20, 6-21
CEX, See Communications Executive
CEXPAR (Communications Executive partition)
  allocating, 3-3
CFE (Configuration File Editor), 6-7
  command references, D-1 to D-13
  exiting, 6-22
  general description, 2-14
  invoking, 6-21
  modifying configuration with, 6-18, 6-20, 6-27
  partition size, B-2
  PEGEN questions, 4-35
  using, 6-22
Checkpointable tasks, 6-8
CLEAR LINE command
  NCP, 6-2, 6-5, 6-7, E-5
  VNP, F-4
CLEAR PROCESS command
  NCP, 6-6, 6-18, E-6
  VNP, F-5
CLEAR SYSTEM command
  NCP, 6-2, 6-8, E-7
  VNP, F-6
Communication errors, 8-2
Communications Executive (CEX), 1-4, 2-1, 2-5, 6-6
  general description, 2-6
  loading, 6-1
  partition (CEXPAR), 3-3
  partition size, B-2
  PEGEN issues, 2-7
  PEGEN questions, 4-28
  processes, 2-7 to 2-11
  system counters, E-19 to E-20
  system parameters (CFE), 6-26 to 6-27
  unloading, 6-2
Communications Executive partition, See CEXPAR
Communications Pool (POOL..), 2-5
  defined, 2-11
  memory allocation, 3-3
Communicatons Control Blocks, See CCBs
Component Build process, A-5, A-9 to A-11

Component parameters (PEGEN), 4-27
Component selection (PEGEN), 4-6, 4-17 to 4-19
Configuration file
  sample listing, A-6 to A-9
Configuration File Editor, See CFE
Control units
  address errors, 8-30
Controllers
  addressing of, 3-12 to 3-16
  modifying address of, 6-27 to 6-29
Counters, line error counters
  DCE, 6-11 to 6-14
  NCP, 6-10 to 6-11
  system, 6-15 to 6-16
Crash dumps, 8-23
CRC (Cyclic Redundancy Check), 2-11
CSR (Control and Status Register)
  addressing, 6-3, 6-4
  modifying address of, 6-23

D

Data Bases, See System image file
Data Link Control characters, 7-3 to 7-10
Database parameters (PEGEN), 4-20 to 4-21
DCE (Device Counters and Errors utility)
  compared with NCP, 8-4
  error counter report, 6-12 to 6-13
  general description, 2-14
  line activity, 6-11 to 6-14
  partition size, B-2
  PEGEN questions, 4-37
DCE Utility, 8-3
DDM (Device Driver Module) processes, 2-9 to 2-11
  PEGEN issues, 2-9
DEFINE LINE command, 6-23
  CFE, D-5 to D-6
DEFINE PROCESS command, 6-25
  CFE, D-7 to D-8
DEFINE SYSTEM command, 6-26 to 6-27
  CFE, D-9 to D-10
Device (interface) problems, 8-31
Device counters utility, See DCE
Device Driver Module, See DDM
Device settings (DV11), C-1 to C-2
Direct Line Access process, See DLX
DLC (Data Link Control) processes
  general description, 2-8
DLE (Data Link Escape)
  defined, 7-4
DLX (Direct Line Access process), 6-7
  error messages, 8-36
  general description, 2-8
  parameters (PEGEN), 4-29
  partition size, B-1
DUP11
  Line loop tests, 8-12
DV device driver, 6-6
  defined, 2-10
  LDBs for, 3-10
  parameters (PEGEN), 4-31
  partition size, B-1
DV11
  Line loop tests, 8-12
DV11 Device settings, C-1 to C-2
DVMOD
  partition size, B-2

E

EBCDIC/ASCII translation, 1-5, 2-8
ENQ (Enquiry)
  defined, 7-7
  functions, 7-9
EOT (End of Transmission)
  defined, 7-7
  functions, 7-9
Equipment Control characters, 7-4
Error handling, 8-1
Error Logging and Statistics Routine, See DCE
Error messages (PE components), 8-33 to 8-36
Error recovery, 8-30
Errors
  counters, 6-11 to 6-14
ETB (End of Transmission Block)
  defined, 7-5
  functions, 7-6
ETX (End of Text)
  defined, 7-6
  functions, 7-6
EXIT command
  NCP, 6-8

H

HELP command
  CFE, D-1
  NCP, E-1
  VNP, F-1

I

I/O, 8-3
IBM 3270 Information Display System (IDS), 1-1
  characteristics emulated by the PE, 1-5
  characteristics emulated by the TE, 1-6 to 1-7
IBM 3271 Control Unit, 1-1

IBM 3271 Protocol Emulation Module. See BSD
IBM Host
  hardware, 1-9 to 1-10
  software, 1-9 to 1-10
IDS. See IBM 3270 Information Display System
Installation errors, 8-1

K

KBU device driver, 6-6
  defined, 2-10
  LDBs for, 3-10
  Parameters (PEGEN), 4-31
  partition size, B-1
KMC11-A microprocessor, 1-4, 1-9, 2-10

L

Large Data Buffers, See LDBs
LBK (Loopback Test Program), 6-7, 8-2
  and KBU lines, 8-11
  and TRA, 8-11
  as part of PE verification, 5-2 to 5-5
  Commands, 8-13 to 8-17
  error messages, 8-37 to 8-39
  general description, 2-15
  HELP command, 8-16
  LOOP command, 8-14 to 8-15
  Modems (Local and Remote), 8-12 to 8-13
  MONITOR command, 8-15 to 8-16
  partition size, B-2
  PEGEN questions, 4-38
  QIO calls, 8-19 to 8-22
  SHOW command, 8-15
  showing status of modem, 5-3
  Using, 8-11 to 8-19
LDBs (Large Data Buffers), 3-7, 3-8
  PEGEN requirements, 3-8 to 3-11
Line counters
  NCP, 6-10 to 6-11
Line errors, 8-3
Line Identification, 2-10 to 2-11
Line interface testing, 5-4 to 5-5
Line loop tests (DUP11 and DV11), 8-12
Line Loopback Test Program, See LBK
Line parameters (PEGEN), 4-22 to 4-25
Line state
  controlling, 6-8
  setting with NCP, 6-3
Lines
  adding, 6-19
  displaying characteristics of (CFE), 6-22 to 6-23
  loading options, 6-3 to 6-5
  modifying parameters (CFE), 6-23 to 6-24
LIST LINE command, 6-22 to 6-23
  CFE, D-11
LIST PROCESS command, 6-24
  CFE, D-12
LIST SYSTEM command, 6-25 to 6-26
  CFE, D-13
LLC (Logical Link Control) processes, 2-7
  PEGEN issues, 2-7
Loading options, 3-5
Log-on errors, 8-30
Logical Link Control processes, See LLC
LOOP command (LBK), 5-4

M

Message Framing Control characters, 7-5 to 7-6
Message Interchange Control characters, 7-6 to 7-10
Mode tables, 2-11
Modem
  monitoring status of (LBK), 5-3
Modem loopback testing, 5-3
Modem problems, 8-31
Modifying component parameters with CFE, 6-22
MONITOR command, 5-4
Multi-Drop line problems, 8-32

N

NAK (Negative Acknowledgement)
  defined, 7-8
  functions, 7-9
  threshold count, 8-2
NCP (Network Control Program), 6-7
  error messages, 8-39 to 8-43
  exiting, E-3
  general description, 2-13
  invoking, E-2
  loading a line with, 6-2 to 6-5
  loading CEX with, 6-1
  loading processes with, 6-5
  modifying configuration with, 6-18, 6-19 to 6-20
  monitoring line activity with, 6-10 to 6-11
  monitoring line status with, 6-9
  monitoring system with, 6-15 to 6-16
  partition layout, 3-6 to 3-7
  partition size, B-2
  PEGEN questions, 4-34
  SHOW LINE COUNTERS, 8-3
  unloading CEX with, 6-2
NDA (Network Dump Analyzer), 8-22 to 8-29
  command strings, 8-24 to 8-26
  error messages, 8-44 to 8-47
  examples, 8-26

general description, 2-15
PEGEN questions, 4-39
switches, 8-27 to 8-29
Network buffer pool, See Buffers
Network Control Program, See NCP
Network Dump Analyzer, See NDA
Network Initializer, See NTINIT
Network Loader, See NTL
NTINIT (Network Initializer), 2-5, 2-12, 6-7
Errors, 8-1
partition layout, 3-6 to 3-7
partition size, B-2
PEGEN questions, 4-32
NTL (Network Loader), 2-5, 2-12, 6-7
error messages, 8-48 to 8-53
Errors, 8-1
partition layout, 3-6 to 3-7
partition size, B-2
PEGEN questions, 4-33

P

Partitions, 3-6
PE.MAC file, 6-27 to 6-29
PECOPY dialog (PEGEN), 4-8 to 4-9
PECOPY procedure (PEGEN), 4-7 to 4-8
PEGEN (PE generation procedure)
and SYSGEN, 3-4 to 3-5
building 3271 PE components, 4-41
general description, 3-1 to 3-2
sample listings, A-1 to A-13
starting, 4-7 to 4-8, 4-10
PEGEN dialog
default for non-responses, 4-3
explanatory text, 4-2
formats for responses, 4-2
general information, 4-1, 4-10
introduction, 4-11 to 4-13
question numbering scheme, 4-4
questions, 4-2
Permanent data base, 2-12
Poll sequence
legal polling addresses for PE, 7-12
message format, 7-11
operation sequence, 7-12 to 7-13
Polling, 3-16, 7-2
POOL.., See Communications Pool
Powerfail recovery, 2-12
Priority controller, 6-23
Procedure parameters (PEGEN), 4-15 to 4-16
Processes
displaying characteristics of (CFE), 6-24
modifying parameters (CFE), 6-25
reloading/renaming partitions, 6-18
Processes (NCP)
changing parameters, 6-5
loading, 6-5
unloading, 6-6
Protocol Emulation (3271)
and communications, 7-1 to 7-2
overview, 7-1 to 7-2
Protocol Emulator (RSX-11/3271)
characteristics, 1-4, 1-5
communications, 1-4
components, 2-5
hardware, 1-7 to 1-9
installing, 5-1 to 5-2
line parameters (PEGEN), 4-26
loading, 5-1 to 5-2
processes, 2-1
reconfiguring with RSX-11M-PLUS, 6-29 to 6-30
software, 1-9
starting, 5-1 to 5-2
tasks, 2-3
verification, 5-2 to 5-5
Pseudodevice Driver, See TUD

R

READ MODIFIED command
defined, 7-13
message format, 7-13
operation sequence, 7-14 to 7-16
Resource Allocation Failure Module, See AUX
RSX-11M-PLUS
reconfiguring with, 6-29 to 6-30
RSX Executive, 2-1
Running data base, 2-12
RVI (Reverse Interrupt)
defined, 7-9
functions, 7-10

S

Screen Emulator
PEGEN questions, 4-40
Selection
defined, 7-18
legal selection addresses for PE, 7-19
message format, 7-18 to 7-19
operation sequence, 7-19 to 7-20
SET KNOWN LINES command
NCP, E-8
VNP, F-7
SET LINE command
NCP, 6-2 to 6-5, 6-7, 6-8, 6-18, E-9 to E-10
VNP, F-8 to F-10
SET PROCESS command
NCP, 6-5, 6-18, E-11 to E-12
VNP, F-11 to F-12

SET SYSTEM command
NCP, E-13
VNP, F-13
SHOW LINE command
NCP, 6-9, E-14
VNP, F-14
SHOW LINE COUNTERS command
NCP, 6-10 to 6-11
SHOW PROCESS command
NCP, E-17 to E-18
VNP, F-15
SHOW SYSTEM command
NCP, 6-15 to 6-17, E-19
VNP, F-16
SOH (Start of Header)
defined, 7-5
functions, 7-6
Stations
addressing, 3-12 to 3-16
modifying address of, 6-27 to 6-29
STATUS AND SENSE READ
defined, 7-16
message format, 7-16 to 7-17
operation sequence, 7-18
status and sense byte definitions, 7-17 to 7-18
STRCRC
partition size, B-2
STX (Start of Text)
defined, 7-5
functions, 7-6
SYN (Synchronous Idle)
defined, 7-4
SYSGEN, 3-2 to 3-5
System
displaying characteristics of, 6-25 to 6-26
System counters, 6-16
System Crashes, 8-23
System image file, 2-12

T

Target system characteristics (PEGEN), 4-14
Terminal Emulator (RSX-11/3271)
characteristics, 1-6 to 1-7
Terminal Unit, 1-4
Terminal Unit Driver Process, See TUD
Topdown loading, 3-6
TRA (Trace program), 8-4 to 8-10
Buffers (size and number), 8-4
commands, 8-5 to 8-6
error messages, 8-54
Example commands, 8-7
Example output, 8-7 to 8-10
general description, 2-14
Output, 8-4
partition size, B-2
PEGEN questions, 4-36
QIOs, 8-4
transmit and receive codes, 8-5
TU units for, 3-16
Trace units, 5-4
Tracing errors (TRA), 8-2
TTB
partition size, B-2
TTD (Temporary Text Delay)
defined, 7-8
functions, 7-10
TU, See Terminal Unit
TUD (Terminal Unit Driver), 1-4, 6-7
general description, 2-8
parameters (PEGEN), 4-29
partition size, B-1

U

UAP, See User Application Programs
UNIBUS Mapping Registers, 3-8, 3-9, 3-11
User Application Programs, 1-2, 1-3

V

Vectors, 6-24
addressing, 6-4
SYSGEN question, 3-4
VNP (Virtual Network program), 6-7
command reference, F-3 to F-16
error messages, 8-54 to 8-59
exiting, F-3
general description, 2-13
invoking, F-2
modifying configuration with, 6-18, 6-19 to 6-20
partition size, B-2
PEGEN questions, 4-41

W

WACK
defined, 7-8
functions, 7-9
WRITE command
defined, 7-21
message format, 7-21 to 7-22
operation sequence, 7-22 to 7-24

Z

ZERO LINE command
NCP, E-21
ZERO SYSTEM command
NCP, E-22

RSX-11 3271 PROTOCOL EMULATOR
System Manager's Guide
AA-C739C-TC

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____ Telephone _____

Street _____

City _____ State _____ Zip Code or Country _____

Cut along dotted line

Do Not Tear-Fold Here and Tape

digital

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SOFTWARE DOCUMENTATION METHODS GROUP, ML3-5/T47
DIGITAL EQUIPMENT CORPORATION,
146 MAIN STREET,
MAYNARD, MASSACHUSETTS 01754

Do Not Tear-Fold Here

Cut Along Dotted Line